



TUGAS AKHIR - SM141501

**IMPLEMENTASI ALGORITMA GENETIKA
UNTUK OPTIMALISASI RANDOM FOREST
DALAM PROSES KLASIFIKASI
PENERIMAAN TENAGA KERJA BARU :
STUDI KASUS PT.XYZ**

**LARAS BINARWATI
NRP 1212 100 089**

**Dosen Pembimbing
Dr. Imam Mukhlash, S.Si, MT
Drs. Soetrisno, M.I.Komp.**

**JURUSAN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2017**



FINAL PROJECT - SM141501

**IMPLEMENTATION GENETIC ALGORITHM FOR
OPTIMIZE RANDOM FOREST IN CLASSIFICATION
PROCESS OF HUMAN RESOURCES
RECRUITMENT : A CASE STUDY IN PT.XYZ**

**LARAS BINARWATI
NRP 1212 100 089**

**Supervisors
Dr. Imam Mukhlash, S.Si, MT
Drs. Soetrisno, M.I.Komp.**

**DEPARTMENT OF MATHEMATICS
Faculty of Mathematics and Natural Science
Sepuluh Nopember Institute of Technology
Surabaya 2017**

LEMBAR PENGESAHAN

IMPLEMENTASI ALGORITMA GENETIKA UNTUK OPTIMALISASI RANDOM FOREST DALAM PROSES KLASIFIKASI PENERIMAAN TENAGA KERJA BARU: STUDI KASUS PT.XYZ

IMPLEMENTATION GENETIC ALGORITHM FOR OPTIMIZE RANDOM FOREST IN CLASSIFICATION PROCESS OF HUMAN RESOURCES RECRUITMENT : A CASE STUDY IN PT.XYZ

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Sains
Pada bidang studi Ilmu Komputer
Program Studi S-1 Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya


Oleh :

LARAS BINARWATI
NRP. 1212 100 089

Menyetujui,

Dosen Pembimbing II,


Dosen Pembimbing I,


Drs. Soetrisno, M.I.Komp.
NIP. 19571103 198603 1 003


Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003

Mengetahui,

Ketua Jurusan Matematika
FMIPA ITS


Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003
Surabaya, 31 Juli 2017

**IMPLEMENTASI ALGORITMA GENETIKA UNTUK
OPTIMALISASI RANDOM FOREST DALAM PROSES
KLASIFIKASI PENERIMAAN TENAGA KERJA BARU: STUDI
KASUS PT.XYZ**

Nama Mahasiswa : Laras Binarwati
NRP : 1212 100 089
Jurusan : Matematika FMIPA-ITS
Dosen Pembimbing : Dr. Imam Mukhlash, S.Si, MT
Drs. Soetrisno, M.I.Komp.

ABSTRAK

Kualitas sumber daya manusia sangat penting bagi suatu perusahaan untuk mempertahankan keunggulan kompetitifnya agar mampu bersaing dengan perusahaan lainnya maupun untuk meningkatkan kualitas dari perusahaan itu sendiri. Oleh karena itu, menggali pola penerimaan tenaga kerja baru sangat diperlukan. Pada penelitian ini, metode *random forest* digunakan untuk menggali pola penerimaan tenaga kerja baru. Adapun algoritma genetika (GA) digunakan untuk mengoptimalkan akurasi berdasarkan pola yang didapat. Hasil pengujian program ini menunjukkan keakuratan pola yang dihasilkan oleh random forest yang dioptimalkan dengan algoritma genetika lebih tinggi dengan hasil keakuratan berkisar antara 91%-95% dibanding dengan hanya menggunakan random forest saja yang hanya berkisar 40-95%.

Kata Kunci : *Penerimaan Tenaga Kerja, Klasifikasi, Random Forest, Algoritma Genetika.*

**IMPLEMENTATION GENETIC ALGORITHM FOR OPTIMIZE
RANDOM FOREST IN CLASSIFICATION PROCESS OF
HUMAN RESOURCES RECRUITMENT :
A CASE STUDY IN PT.XYZ**

Name	:	Laras Binarwati
NRP	:	1212 100 089
Department	:	Mathematics FMIPA-ITS
Supervisors	:	Dr. Imam Mukhlash, S.Si, MT Drs. Soetrisno, M.I.Komp.

ABSTRACT

The quality of human resources is very important for a company to maintain its competitive advantage in order to compete with other companies and also to improve the quality of the company itself. Therefore, discovering new labor recruitment patterns is necessary. In this research, the random forest method is used to discover new recruitment patterns. The genetic algorithm (GA) is used to optimize the accuracy based on the obtained patterns. The results of this program show the accuracy of pattern generated by optimized random forest with genetic algorithm ranging between 91% -95%, that result is higher than using only random forest method thus ranging between 40-95%.

Keyword : *Labor Recruitement, Classification, Random Forest, Genetic Algorithm*

KATA PENGANTAR

Puji syukur penulis panjatkan pada kehadiran Allah Swt, karena hanya dengan karunia rahmat, bimbingan, serta anugrah-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul

“IMPLEMENTASI ALGORITMA GENETIKA UNTUK OPTIMALISASI RANDOM FOREST DALAM PROSES KLASIFIKASI PENERIMAAN TENAGA KERJA BARU: STUDI KASUS PT.XYZ”

yang merupakan salah satu persyaratan akademik dalam menyelesaikan Program Sarjana Departemen Matematika, Institut Teknologi Sepuluh Nopember Surabaya.

Dalam proses pembuatan Tugas Akhir ini, penulis mendapat bantuan dan dukungan dari berbagai pihak. Untuk itu penulis mengucapkan terima kasih kepada :

1. Ketua Departemen Matematika FMIPA-ITS yang telah memberi dukungan dan kemudahan pengurusan persyaratan-persyaratan selama penulis menyelesaikan Tugas Akhir ini.
2. Bapak Dr. Imam Mukhlash, S.Si, MT sebagai dosen pembimbing I Tugas Akhir atas segala bimbingan dan motivasi yang telah diberikan pada penulis.
3. Bapak Drs. Soetrisno, M.I.Komp. selaku pembimbing II yang senantiasa membimbing dan mengarahkan dalam penyusunan Tugas Akhir ini
4. Bapak Drs. Kamiran, M.Si, Dra. Wahyu Fistia Doctorina, M.Si, Drs. Nurul Hidayat, M.Kom., selaku dosen penguji atas semua saran yang telah diberikan untuk perbaikan Tugas Akhir ini.
5. Dr. Didik Khusnul Arif, S.Si., M.Si dan Drs. Iis Herisman, M.Sc. selaku selaku Kaprodi dan Sekretaris Kaprodi Departemen Matematika yang mengkoordinir jadwal Tugas Akhir.

6. Bapak Drs. Soehardjoepri, M.Si selaku dosen wali penulis yang telah memberikan motivasi dan arahan akademik.
7. Bapak dan Ibu Dosen serta seluruh staf Tata Usaha dan Laboratorium Departemen Matematika FMIPA-ITS.
8. Rizka Fika Asanul In'am selaku narasumber yang telah bersedia meluangkan waktu untuk memberikan informasi yang dibutuhkan penulis untuk menyelesaikan Tugas Akhir.
9. Keluarga tercinta terutama Bapak Nurcholis dan Ibu Agustina, penulis ucapkan banyak terima kasih atas doa serta dukungan yang telah diberikan baik moral maupun material, serta Guntur Perdana Lahutama dan Bias Merdeka Lahutama yang telah memberikan semangat dalam penyusunan Tugas Akhir ini.
10. Sahabat – sahabat penulis Lena, Sheerty, Maya, Firda, dan Ditta yang telah memberikan semangat dan hiburan serta sebagai tempat berbagi apapun.
11. Teman – teman seperjuangan Matematika ITS 2012 khususnya MAT12IKS tercinta, dan kawan-kawan yang telah banyak membantu baik secara langsung maupun tidak.
12. Sahabat penulis sedari SMA Zahra, Niken, Laksmi, Yosevina, Dayinta, Didi, dan Ima yang telah banyak memberikan saran dan motivasi.
13. Syafriandi Nurrahman yang senantiasa mendampingi, mengingatkan serta saling mendoakan untuk tetap semangat sampai akhir.
14. Seluruh pihak yang tidak dapat penulis sebutkan satu persatu, yang turut membantu dalam penyusunan Tugas Akhir ini.

Penulis menyadari bahwa selama masa penelitian dan penyusunan laporan ini masih banyak kekurangan dan kesalahan. Oleh karena itu, penulis memohon saran dan kritik sebagai bahan perbaikan di masa yang akan datang. Semoga Tugas Akhir ini bermanfaat bagi semua pihak.

Surabaya, Juli 2017

Penulis

DAFTAR ISI

JUDUL.....	i
LEMBAR PENGESAHAN.....	v
ABSTRAK.....	vii
<i>ABSTRACT</i>	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL.....	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	4
1.5 Manfaat.....	5
1.6 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA	7
2.1 Penelitian Terdahulu.....	7
2.2 <i>Data Mining</i>	14
2.3 <i>Knowledge Discovery in Database (KDD)</i>	15
2.4 <i>Classification</i>	16
2.5 <i>Random Forest</i>	18
2.6 Algoritma Genetika.....	19
BAB III METODE PENELITIAN	21
3.1 Studi Literatur.....	21
3.2 Pengumpulan Data.....	21
3.3 Pengolahan Data.....	22
3.4 Perancangan <i>Interface</i> dan Evaluasi Sistem.....	24

3.5	Evaluasi dan Kesimpulan.....	24
BAB IV ANALISIS DAN PERANCANGAN SISTEM		27
4.1	Gambaran Umum Perangkat Lunak.....	27
4.2	<i>Usecase</i> Diagram.....	28
4.3	<i>Acitivity</i> Diagram.....	29
4.4	Pendefinisian Individu.....	30
4.5	Inisialisasi Kromosom.....	30
4.6	<i>Random Forest</i>	30
4.7	<i>Crossover</i> pada Algoritma Genetika.....	39
4.8	Mutasi pada Algoritma Genetika.....	41
4.9	Desain <i>Interface</i>	43
BAB V PENGUJIAN DAN PEMBAHASAN		45
5.1	<i>Pre-Processing</i> dan Pembagian Data Uji.....	45
5.2	Representasi Data.....	46
5.3	Inisialisasi Kromosom.....	47
5.4	Akurasi <i>Random Forest</i>	48
5.5	Akurasi Algoritma Genetika.....	50
BAB VI PENUTUP		53
6.1	Kesimpulan.....	53
6.2	Saran.....	53
DAFTAR PUSTAKA.....		55
LAMPIRAN A Data Karyawan.....		57
LAMPIRAN B Data Arff.....		75
LAMPIRAN C Source Code.....		77
LAMPIRAN D Hasil Uji Coba.....		97
LAMPIRAN E Biodata Penulis.....		109

DAFTAR GAMBAR

Gambar 2.1	Konsep Analisis <i>Framework</i>	11
Gambar 2.2	RFGA <i>Workflow</i>	14
Gambar 2.3	Proses KDD di <i>Database</i>	16
Gambar 3.1	Diagram Alur Penelitian.....	25
Gambar 4.1	<i>Usecase</i> Diagram Sistem Aplikasi.....	28
Gambar 4.2	<i>Activity</i> Diagram Sistem Algoritma Genetika Untuk Optimalisasi Data Dalam proses Klasifikasi Penerimaan tenaga Kerja Baru Menggunakan Metode <i>Random</i> <i>Forest</i>	29
Gambar 4.3	Alur Kerja <i>Random Forest</i>	31
Gambar 4.4	Struktur <i>Random Forest</i>	32
Gambar 4.5	Struktur <i>Random Forest</i>	35
Gambar 4.6	Prediksi <i>Random Forest</i>	36
Gambar 4.7	Prediksi <i>Random Forest</i>	37
Gambar 4.8	Alur Kerja <i>Crossover</i>	40
Gambar 4.9	Form Utama.....	43
Gambar 5.1	Hasil Tampilan Data sebelum <i>Pre- Processing</i>	45
Gambar 5.2	Hasil <i>Pre-processing</i>	46
Gambar 5.3	Hasil <i>Pre-Processing</i>	46
Gambar 5.4	Tampilan Awal Sebelum Input Jumlah Inisialisasi.....	47
Gambar 5.5	Input Jumlah Inisialisasi.....	47
Gambar 5.6	Hasil Akurasi <i>Random Forest</i>	48
Gambar 5.7	Hasil Akurasi Setelah di Optimalisasi dengan Algoritma Genetika.....	50

DAFTAR TABEL

Tabel 2.1	Konsep Analisis <i>Framework</i>	7
Tabel 2.2	Konsep Analisis <i>Framework</i>	8
Tabel 2.3	Contoh Rules Job Performance.....	12
Tabel 2.4	Contoh Rules Retensi.....	12
Tabel 2.5	Frekuensi Setiap Atribut.....	13
Tabel 4.1	Data Awal.....	34
Tabel 4.2	Data Resample.....	34
Tabel 4.3	Hasil Prediksi.....	38

BAB I

PENDAHULUAN

Pada bab ini akan dijelaskan mengenai latar belakang dari permasalahan yang dibahas pada Tugas Akhir, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan dari Tugas Akhir ini.

1.1 Latar Belakang Masalah

Akhir-akhir ini penelitian mengenai perekrutan karyawan telah meningkat secara dramatis. Kualitas sumber daya manusia sangat penting bagi suatu perusahaan untuk mempertahankan keunggulan kompetitifnya agar mampu bersaing dengan perusahaan lainnya maupun untuk meningkatkan kualitas dari perusahaan itu sendiri. Banyak penelitian awal difokuskan pada penggunaan ulasan pekerjaan yang realistis (yaitu, memberikan para pelamar kerja dengan informasi yang akurat tentang posisi dalam organisasi), metode perekrutan tradisional (misalnya, iklan pada koran), dan karakteristik para pelamar (misalnya, perilaku mereka), maka dari itu rekrutmen personal dan seleksi secara langsung merupakan cara yang dianggap sangat mempengaruhi dari segi kualitas karyawan tersebut. Pemilihan calon karyawan yang tepat adalah salah satu faktor yang paling penting untuk keberhasilan keputusan strategis jangka panjang yang diambil di dalam suatu perusahaan.

Namun, karena sifat perubahan pengetahuan pekerja di suatu perusahaan, pekerjaan tidak dapat dengan mudah digambarkan terutama untuk pekerjaan di tingkat manajemen. Persyaratan kualitas personil di suatu perusahaan semakin ketat, sedangkan proses kerja di perusahaan itu menjadi beragam dan rumit. Dengan demikian, pendekatan konvensional dengan seleksi personal yang dikembangkan atas dasar karakteristik pekerjaan statis tidak akan lagi cukup [11]. Dalam rangka untuk mencari orang yang tepat untuk melakukan hal yang benar untuk pekerjaan yang tepat,

mengembangkan pendekatan pilihan yang efektif sangatlah penting. Salah satu hal penting dalam menentukan tindakan preventif ini dengan mengetahui pola yang telah terjadi sebelumnya. Salah satu cara yang dapat dilakukan untuk membuat pola tersebut dapat dilakukan dengan data mining.

Data mining merujuk pada ekstraksi penggunaan pola atau aturan dari database yang besar melalui eksplorasi dan analisis data otomatis atau semi-otomatis [12]. Teknik data mining telah banyak diterapkan di berbagai bidang dan telah membuahkan hasil yang luar biasa. Penerapan data mining dalam menganalisis pola untuk penerimaan tenaga kerja diharapkan mampu menggali informasi dan menganalisis hasilnya untuk menemukan tenaga kerja yang tepat dan berkualitas yang dapat dilakukan oleh pihak perusahaan. *Classification* merupakan salah satu metode yang dapat digunakan untuk menemukan pola tersebut. Dari pola yang didapatkan, kita dapat memprediksi dan melihat gambaran untuk bisa membantu menentukan penempatan tenaga kerja tersebut sehingga pihak perusahaan dapat mengambil tindakan yang tepat. *Classification* yang dapat digunakan untuk menentukan pola penerimaan tenaga kerja ini bekerja dengan menentukan variabel penerimaan tenaga kerja baru. Ketepatan dalam penentuan pola ini sangat penting saat mengambil keputusan yang dipilih.

C.-F. Chien dan L.-F. Chen telah melakukan penelitian mengenai penjelasan dan prediksi pola penerimaan tenaga kerja pada tahun 2008[5]. Penelitian ini bertujuan untuk memprediksi dan menentukan dengan mengembangkan kerangka data mining berdasarkan pohon keputusan dan asosiasi rule untuk menghasilkan keputusan penerimaan tenaga kerja baru. Studi kasus yang digunakan dalam penelitian ini adalah sebuah perusahaan di bidang teknologi industri tingkat tinggi. Variabel yang dipakai sebanyak 4 macam yang berasal dari faktor internal karyawan dan sistem perusahaan itu sendiri. Hasil dari penelitian ini menunjukkan bahwa metode decision tree dengan *classification Algorithm* dapat mempermudah dan dapat membantu mengambil suatu keputusan untuk penerimaan tenaga kerja baru.

Pada penelitian lainnya membahas tentang pengaplikasian algoritma genetik (GA) untuk mengoptimalkan random forest pada data kelas rekayasa[4]. Hasil membuktikan keunggulan metode yang diusulkan dalam menaikkan akurasi. Kelengkapan training data set (contoh data pelatihan) memegang peran penting dalam menentukan kualitas pengklasifikasi pola yang dihasilkan. Penelitian ini menyarankan tiga komponen sistem untuk meningkatkan akurasi klasifikasi dalam random forests. Komponen pertama adalah kelas dekomposisi, kedua adalah dengan menggunakan dua parameter, yaitu jumlah tree dalam ensemble, dan jumlah sampel acak yang menonjol pada setiap node split di setiap tree. Ketiga adalah penggunaan algoritma genetika. Penelitian ini menggunakan 22 data set, dan hasilnya membuktikan keefektifan teknik pembelajaran mesin hybrid yang diusulkan dalam meningkatkan akurasi prediksi.

Algoritma genetika merupakan salah satu algoritma yang bersifat heuristic dan dapat menyelesaikan permasalahan multi objektif, sehingga dapat diterapkan untuk mencari solusi optimal dari permasalahan optimasi. Algoritma genetika banyak digunakan dalam masalah optimasi dan mempunyai kemampuan untuk menghasilkan solusi yang baik untuk masalah-masalah rumit [8].

Data set untuk penelitian ini diambil dari perusahaan yang bergerak dibidang pelayaran, yang salah satu fungsinya sebagai Transportir pemuat Cargo bijih besi maupun batubara. Kualitas sumber daya manusia sangat penting bagi perusahaan untuk mempertahankan keunggulan kompetitif dalam era sekarang ini. Namun, perusahaan sering merasa sulit untuk merekrut bakat yang tepat. Pada Tugas Akhir ini akan dibangun suatu perangkat lunak yang mampu menggali pola penerimaan tenaga kerja baru menggunakan random forest yang dioptimalkan dengan menggunakan algoritma genetika.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, permasalahan dalam Tugas Akhir ini dapat disusun sebagai berikut:

1. Bagaimana menggali pola penerimaan tenaga kerja pada suatu perusahaan menggunakan data mining dengan metode *random forests*?
2. Bagaimana perbandingan keakuratan antara hasil yang diberikan oleh random forest dan yang telah dioptimalkan dengan algoritma genetika?
3. Apakah pola yang dihasilkan dari penggunaan *random forests* yang dioptimalkan dengan menggunakan algoritma genetika dapat membantu dalam pengambilan keputusan penerimaan tenaga kerja baru?

1.3 Batasan Masalah

Pada penelitian ini, permasalahan akan dibatasi pada hal-hal sebagai berikut:

1. Objek yang digunakan pada penelitian ini adalah 7 variabel yang berasal dari data karyawan dan perilaku dalam bekerja untuk menentukan pola yang tepat.
2. Asumsi data sudah baik dan tidak membahas proses warehouse.
3. Data yang digunakan pada penelitian ini adalah data dari perusahaan yang bergerak dibidang pelayaran shipping company, yang salah satu fungsinya sebagai Transportir pemuat Cargo bijih besi maupun batubara.

1.4 Tujuan

Berdasarkan permasalahan yang telah dirumuskan sebelumnya, tujuan penelitian Tugas Akhir ini adalah implementasi random forests dan mengoptimalkannya dengan

menggunakan algoritma genetika dalam perangkat lunak untuk mendapatkan pola penerimaan tenaga kerja baru sehingga dapat membantu dalam pengambilan keputusan.

1.5 Manfaat

Tugas Akhir ini dapat memberikan manfaat sebagai berikut:

1. Diperoleh suatu metode untuk membantu dalam pengambilan suatu keputusan penerimaan tenaga kerja pada perusahaan yang menjadi studi kasus.
2. Sebagai salah satu referensi penggunaan *classification* dalam penggalian pola dengan *random forests* yang telah di optimalkan dengan menggunakan algoritma genetika.

1.6 Sistematika Penulisan

Sistematika dari penulisan penelitian ini adalah sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini berisi tentang gambaran umum dari penulisan Tugas Akhir ini yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan, manfaat penelitian, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Pada bab ini menjelaskan tentang kajian teori dari referensi penunjang serta penjelasan permasalahan yang dibahas dalam Tugas Akhir ini, meliputi penelitian sebelumnya terkait tugas akhir ini, pengertian *Data Mining*, *Knowledge Discovery in Database (KDD)*, *Classification*, *Decision Tree Algorithm*, *Random Forests*, dan Algoritma Genetika.

3. BAB III METODOLOGI PENELITIAN

Pada bab ini dibahas tentang langkah – langkah dan metode

yang digunakan untuk menyelesaikan Tugas Akhir ini.

4. **BAB IV ANALISIS DAN PERANCANGAN SISTEM**

Pada bab ini akan menguraikan bagaimana tahapan tahapan dalam analisis dan perancangan sistem. Pembahasan analisis sistem dimulai dari deskripsi perangkat lunak hingga pemodelan analisis sistem, sedangkan perancangan sistem dimulai dari perancangan perangkat lunak hingga perancangan proses.

5. **BAB V HASIL DAN PEMBAHASAN PENGUJIAN**

Bab ini menjelaskan tentang implementasi rancangan perangkat lunak dalam bahasa pemrograman Java. Hasil uji coba perangkat lunak tersebut menggunakan data penelitian yang sudah ada.

6. **BAB VI PENUTUP**

Bab ini berisi kesimpulan yang diperoleh dari pembahasan masalah sebelumnya serta saran yang diberikan untuk pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan mengenai tinjauan pustaka yang menjadi dasar materi dalam penyusunan Tugas Akhir serta menunjang metode – metode yang digunakan dalam pembahasan Tugas Akhir ini.

2.1 Penelitian Terdahulu

Akhir - akhir ini penelitian mengenai aplikasi sumber daya manusia telah banyak dilakukan. Salah satu pengembangan dari topik ini adalah mengenai penerimaan tenaga kerja baru di suatu perusahaan, terutama untuk memastikan sumber daya manusia tersebut adalah orang yang tepat untuk pekerjaan yang tepat dan diwaktu yang tepat. Dalam data mining, data mengenai sumber daya manusia yang dikhususkan pada penerimaan tenaga kerja baru telah banyak digunakan sebagai objek penelitian.

Hamidah Jantan, Abdul Razak Hamdan, dan Zulaiha Ali Othman membuat sebuah penelitian untuk membandingkan teknik data mining apa yang terbaik untuk digunakan pada pengaplikasian sumber daya manusia [2]. Tabel 2.1 merupakan hasil perbandingan dari teknik data mining pada aplikasi sumber daya manusia

Tabel 2.1. Konsep analisis *framework*

Metode Data Mining	Manajemen Sumber Daya Manusia
<i>Fuzzy Data Mining dan Fuzzy Artificial Neural Network</i>	Pengembangan karyawan - penugasan
<i>Decision tree</i>	Seleksi personil -

	Sikap kerja
<i>Association rule mining</i>	Pengembangan karyawan – Pelatihan
<i>Rough Set Theory</i>	Seleksi Personil – Merekrut dan Mempertahankan
<i>Fuzzy Data Mining</i>	Seleksi Personil

Terlihat pada Tabel 2.1 bahwa teknik terbaik adalah menggunakan *Decision tree*, *Rough set theory*, dan *fuzzy data mining*. Faktanya, *Decision tree* memiliki keuntungan mudah untuk dipahami bagi para pengambil keputusan untuk membandingkan dengan pengetahuan domain mereka untuk validasi dan membenarkan keputusan mereka[6]. Selain itu, *Decision tree* dapat menganalisis berbagai data tanpa memerlukan asumsi tentang distribusi yang mendasari. Keuntungan menggunakan *Decision tree* dapat dilihat pada Tabel 2.2

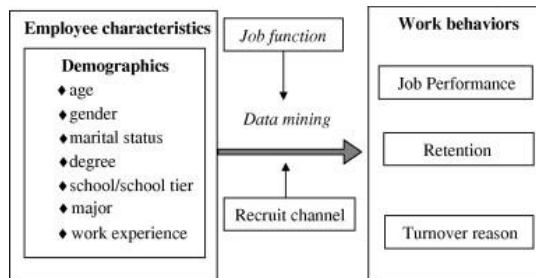
Tabel 2.2. Konsep analisis *framework*

Teknik Data Mining	Karakteristik
<i>Artificial Neural Network</i>	Menyediakan berbagai alat canggih untuk optimasi, aproksimasi fungsi, pemanggilan pola dan pemodelan
<i>Decision Tree</i>	<ol style="list-style-type: none"> 1. Biasanya digunakan untuk tugas klasifikasi dan prediksi 2. Menghasilkan sebuah model yang dapat mewakili aturan interpretasi atau pernyataan logika

	<p>3. Non-parametrik: cocok untuk menangkap bentuk fungsional yang berkaitan dengan variabel independen dan dependen</p> <p>4. Mudah ditafsirkan, komputasi murah dan mampu menangani noisy data. Prediksi model dapat dijelaskan oleh model pengguna</p> <p>5. Deteksi interaksi otomatis - temukan dengan cepat signifikansi interaksi orde tinggi</p> <p>6. Output lebih informatif</p>
<i>Rough Set Theory</i>	Dapat menjelaskan dan mengeksplorasi bagaimana keputusan dibuat dengan aturan sederhana, mudah dimengerti, dan berguna di ketidakpastian dan ketidakjelasan
<i>Fuzzy Clustering</i>	<p>Lebih umum daripada metode konvensional.</p> <p>Digunakan untuk membangun hubungan antar data dan mentransformasikan hubungan menjadi pengetahuan</p>

<i>SVM</i>	Dapat digunakan untuk prediksi serta klasifikasi dan memberikan deskripsi ringkas tentang model yang dipelajari dan sangat akurat
------------	---

Chen-Fu Chien dan Li-Fie Chen mengangkat topik mengenai penerimaan tenaga kerja baru menggunakan *data mining* [5]. Mereka melakukan penelitian menggunakan *decision tree* dengan algortima CHAID. Variabel yang digunakan sebanyak 14 variabel yang dibagi kedalam 4 kategori, yaitu performa kerja (n=3), alasan pengunduran diri (n=4), data diri (n=8), dan lama bekerja. Peneliti ini terdiri dari 2 tahap, yaitu inialisasi variabel yang paling relevan dan menghitung kapasitas variabel tersebut untuk memprediksi pola penerimaan tenaga kerja baru. Untuk mengidentifikasi penerimaan tenaga kerja baru yang efektif untuk mendapatkan bakat berpotensi tinggi yang tepat untuk fungsi pekerjaan yang berbeda, penelitian ini mengembangkan kerangka *data mining* untuk menganalisis data sumber daya manusia, di mana pohon keputusan dipekerjakan untuk mengekstrak rules antara profil pelamar dan perilaku kerja mereka . Dengan kata lain, tujuan dari penelitian ini adalah untuk memprediksi perilaku kerja pelamar termasuk kinerja dan retensi berdasarkan masukan dari atribut profil yang diperoleh di tahap seleksi. variabel input ini termasuk data demografi seperti usia, jenis kelamin, status perkawinan, latar belakang pendidikan, pengalaman kerja dan saluran rekrutmen seperti internal atau eksternal. Gambar 2.1 menggambarkan struktur konseptual.



Gambar 2.1 Konsep analisis *framework* [5]

Data yang digunakan untuk menganalisis adalah data pada karyawan baru yang dipekerjakan dari tahun 2001-2004 untuk 19 fungsi pekerjaan. Ukuran sampel total sebanyak 5289. Untuk analisis kinerja, terfokus pada penggalian informasi yang berkaitan dengan orang-orang yang dengan status kerja "luar biasa" (10%) atau "butuh perbaikan" (bawah 5%). Adapun analisis retensi, 940 kasus dari 5289 sampel telah berhenti dari pekerjaan mereka. Menurut pengalaman ahli sumber daya manusia dalam perusahaan, jika salah satu karyawan berhenti bekerja dalam masa percobaan tiga bulan, proses rekrutmen dianggap gagal dan investasi pelatihan menjadi terbuang. Di sisi lain, jika seorang karyawan berhenti bekerja pekerjaan dalam waktu satu tahun setelah ia dipekerjakan, itu dianggap sebagai masalah manajemen. Dengan demikian, karakteristik mereka yang gagal dalam masa percobaan dan mereka yang berhenti dalam waktu satu tahun secara khusus dianalisis untuk mengungkap hubungan profil pribadi dan fungsi pekerjaan.

Setelah dianalisis, 50 rules yang terkait dengan kinerja dan 16 rules yang terkait dengan retensi ditemukan. Tabel 2.3 dan tabel 2.4 adalah daftar beberapa aturan yang diturunkan. Tabel 2.5 menunjukkan frekuensi setiap atribut yang terhubung ke prediksi target. Seperti yang ditunjukkan dalam hasil, variabel fungsi,

tingkatan sekolah, derajat, dan pengalaman merupakan atribut utama yang berhubungan dengan target yang diprediksi.

Tabel 2.3. Contoh *rules job performance*

3	IF degree = others	1.14	5
	THEN he/she will perform with a level of improvement needed. ($n = 105$; outstanding performance rate = 90%)		
4	IF school tier = {2,4}	1.09	5
	THEN he/she will perform with a level of improvement needed. ($n = 149$; confidence = 86%)		
5	IF degree = master's and above, recruitment channel = internal, and school tier = {1,2}	3.00	
	THEN he/she will perform excellently. ($n = 24$; confidence = 63%)		
6	IF degree = master's and above, recruitment channel = external, and school tier = {2,3,4}	1.12	
	THEN he/she will perform with a level of improvement needed. ($n = 27$; confidence = 96%)		

Tabel 2.4. Contoh *rules retensi*

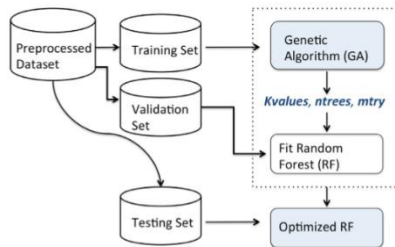
Some examples of the derived rules for resignation

No.	Rule	Lift
1	IF function = C and experience = yes	2.52
	THEN he/she will quit within three months. ($n = 94$; resignation rate = 20%)	
2	IF function = C and experience = no	1.02
	THEN he/she will not quit within three months. ($n = 245$; resignation rate = 94%)	
3	IF function = B and recruit channel = internal	1.15
	THEN he/she will quit within one year. ($n = 75$; resignation rate = 31%)	
4	IF function = B and recruit channel = external	1.08
	THEN he/she will not quit within three months. ($n = 208$; resignation rate = 79%)	
5	IF function = C and experience = yes and recruit channel = external	4.65
	THEN he/she will quit within three months. ($n = 27$; resignation rate = 37%)	

Tabel 2.5.Frekuensi setiap atribut yang terkait dengan prediksi target

Target	Attribut						
	Functions	School	Recruitment channel	Degree	Experience	Job Performance	Major
Job Performance	14	26	26	22	20	0	2
Retention	22	8	8	4	4	6	0
Total	36	34	34	26	24	6	2

Penelitian mengenai pengoptimalan random forests menggunakan metode algoritma genetika pernah dilakukan oleh Eyad Elyan dan Mohamed Medhat Gaber [4]. Penelitian ini menunjukkan hasil dari pengoptimalan metode *random forests*. Hasil membuktikan keunggulan metode yang diusulkan dalam menaikkan akurasi. Kelengkapan *training data set* (contoh data pelatihan) memegang peran penting dalam menentukan kualitas pengklasifikasi pola yang dihasilkan. Penelitian ini menyarankan tiga komponen sistem untuk meningkatkan akurasi klasifikasi dalam *random forests*. Komponen pertama adalah *class decomposition*, kedua adalah dengan menggunakan dua parameter, yaitu jumlah *tree* dalam ensemble, dan jumlah sampel acak yang menonjol pada setiap node split di setiap *tree*. Ketiga adalah penggunaan algoritma genetika. Penelitian ini menggunakan 22 data set, dan hasilnya membuktikan effectiveness teknik pembelajaran mesin hybrid yang diusulkan dalam meningkatkan accuracy prediktif.



Gambar 2.2 Workflow Random Forests Genetic Algoritma (RFGA) [4]

Gambar 2.2 menjelaskan alur kerja dari algoritma genetika pada random forests. Perlu diperhatikan bahwa data training telah digunakan pada saat proses pengoptimalan selama data valid digunakan untuk uji coba optimasi random forests proses training. Data testing digunakan hanya pada saat mengkaji hasil. Dengan kata lain, data testing hanya digunakan ketika proses hasil akhir, terutama untuk uji coba hasil optimasi model random forests.

2.2 Data Mining

Data mining merupakan proses ekstraksi pola yang penting dari data dalam jumlah besar. *Data mining* merupakan salah satu langkah dalam proses *Knowledge Discovery in Database (KDD)* untuk menemukan pola yang bermanfaat. *Data mining* juga didefinisikan sebagai suatu proses yang menggunakan berbagai perangkat analisis data untuk menemukan pola dan relasi data agar dapat digunakan untuk membuat prediksi dengan tepat. Terdapat dua kunci sukses dalam *data mining*, yaitu ketepatan formulasi masalah yang akan dipecahkan dan penggunaan data yang tepat. Setelah mendapatkan data yang akan digunakan untuk dianalisis,

ketepatan dalam mentransformasikan dan mengolahnya sangat mempengaruhi ketepatan pengetahuan hasil penggalian data [3].

Beberapa karakteristik umum yang dimiliki data yang akan dianalisis adalah sebagai berikut [4]:

1. Ukuran Data Besar
2. Data tidak lengkap, sehingga perlu proses *cleaning*
3. Struktur data kompleks
4. Merupakan data yang heterogen.

2.3 Knowledge Discovery in Database (KDD)

Knowledge Discovery in Database (KDD) adalah proses menemukan informasi yang bermanfaat serta pola-pola yang ada dalam data. KDD merupakan sebuah proses yang terdiri dari serangkaian proses iteratif yang terurut dan *data mining* merupakan salah satu langkah dalam KDD. Ada Beberapa langkah yang dilakukan dalam KDD [3]:

1. Pembersihan Data
Pembersihan data dilakukan untuk menghilangkan data yang tidak konsisten atau mengandung *noise*.
2. Integrasi Data
Proses integrasi data dilakukan untuk menggabungkan data dari berbagai basis data.
3. Seleksi Data
Proses seleksi data dilakukan dengan mengambil data yang relevan untuk proses analisis.
4. Transformasi Data
Proses ini dilakukan untuk mentransformasikan atau menggabungkan data ke dalam bentuk yang tepat agar proses *data mining* dapat dilakukan dengan lebih mudah.
5. *Data Mining*
Data Mining merupakan proses penting dimana metode-metode tertentu diaplikasikan untuk mengekstrak pola-pola unik dalam data.

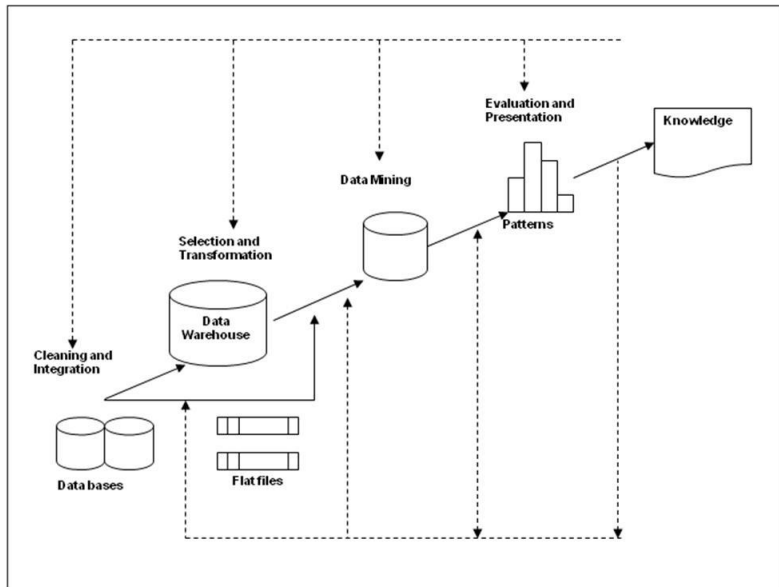
6. Evaluasi Pola

Evaluasi pola diperlukan untuk mengidentifikasi pola-pola menarik yang ditemukan sebagai bahan dalam representasi pengetahuan.

7. Presentasi Pengetahuan

Penggunaan visualisasi dan teknik representasi untuk menunjukkan pengetahuan hasil penggalian data kepada pengguna.

Tahapan-tahapan proses KDD secara berurut dapat dilihat pada Gambar 2.3 berikut ini:



Gambar 2.3 Proses Knowledge Discovery in Database

2.4 Classification

Classification adalah proses penemuan model (fungsi) yang menggambarkan dan membedakan kelas data atau konsep yang

bertujuan agar bisa digunakan untuk memprediksi kelas dari objek yang label kelasnya tidak diketahui [3].

Classification data terdiri dari 2 langkah proses. Pertama adalah *learning* (fase *training*), dimana algoritma *classification* dibuat untuk menganalisa data *training* lalu direpresentasikan dalam bentuk *rule classification*. Proses kedua adalah *classification*, dimana data tes digunakan untuk memperkirakan akurasi dari *rule classification* [5].

Proses *classification* didasarkan pada empat komponen [1]:

- a. *Kelas* . Variabel dependen yang berupa kategorikal yang merepresentasikan “label” yang terdapat pada objek. Contohnya: resiko penyakit jantung, resiko kredit, *customer loyalty*, jenis gempa.
- b. *Predictor* . Variabel independen yang direpresentasikan oleh karakteristik (atribut) data. Contohnya: merokok, minum alkohol, tekanan darah, tabungan, aset, gaji.
- c. *Training dataset*. Satu set data yang berisi nilai dari kedua komponen di atas yang digunakan untuk menentukan kelas yang cocok berdasarkan *predictor*.
- d. *Testing dataset*. Berisi data baru yang akan diklasifikasikan oleh model yang telah dibuat dan akurasi klasifikasi dievaluasi.

Berikut ini adalah algoritma *classification* dalam *data mining* yang paling populer yaitu adalah [1]:

- a. *Decision/classification trees*
- b. *Bayesian classifiers/Naive Bayes classifiers*
- c. *Neural networks*
- d. *Statistical analysis*
- e. *Rough sets*
- f. *k-nearest neighbor classifier*
- g. *Rule-based methods*
- h. *Memory based reasoning*
- i. *Support vector machines*.

2.5 *Random Forests*

Metode *random forest* adalah pengembangan dari metode CART, yaitu dengan menerapkan metode *bootstrap aggregating* (bagging) dan *random feature selection* [13]. Dalam *random forest*, banyak pohon ditumbuhkan sehingga terbentuk hutan (*forest*), kemudian analisis dilakukan pada kumpulan pohon tersebut. Pada gugus data yang terdiri atas n amatan dan p peubah penjelas, *random forest* dilakukan dengan cara [13]:

1. lakukan pengambilan contoh acak berukuran n dengan pemulihan pada gugus data. Tahapan ini merupakan tahapan bootstrap
2. dengan menggunakan contoh bootstrap, pohon dibangun sampai mencapai ukuran maksimum (tanpa pemangkasan). Pada setiap simpul, pemilihan pemilah dilakukan dengan memilih m peubah penjelas secara acak, dimana $m \ll p$. Pemilah terbaik dipilih dari m peubah penjelas tersebut. Tahapan ini adalah tahapan random feature selection
3. Ulangi langkah 1 dan 2 sebanyak k kali, sehingga terbentuk sebuah hutan yang terdiri atas k pohon

Respons suatu amatan diprediksi dengan menggabungkan (aggregating) hasil prediksi k pohon. Pada masalah klasifikasi dilakukan berdasarkan majority vote (suara terbanyak).

Error klasifikasi random forest diduga melalui error OOB yang diperoleh dengan cara [13]:

1. Lakukan prediksi terhadap setiap data OOB pada pohon yang bersesuaian. Data OOB (out of bag) adalah data yang tidak termuat dalam contoh bootstrap.
2. Secara rata-rata, setiap amatan gugus data asli akan menjadi data OOB sebanyak sekitar 36% dari banyak pohon. Oleh

karena itu, pada langkah 1, masing-masing amatan gugus data asli mengalami prediksi sebanyak sekitar sepertiga kali dari banyaknya pohon. Jika a adalah sebuah amatan dari gugus data asli, maka hasil prediksi random forest terhadap a adalah gabungan dari hasil prediksi setiap kali a menjadi data OOB.

3. Error OOB dihitung dari proporsi misklasifikasi hasil prediksi random forest dari seluruh amatan gugus data asli.

2.6 Algoritma Genetika

Algoritma Genetika atau Genethic Algorithm (GA) dikenalkan oleh John Holland dalam menyelesaikan masalah optimasi. Algoritma Genetika mensimulasikan proses yang terjadi pada populasi alamiah yang merupakan hal yang penting dalam proses evolusi. Algoritma Genetika adalah metode pencarian yang meniru perumpamaan evolusi biologis alami untuk menentukan kromosom atau individu berkualitas tinggi dalam suatu kawasan berhingga potensial yang disebut populasi. Proses pemilihan individu dari suatu populasi dievaluasi berdasarkan fungsi fitness. Kromosom berwujud string tersebut merupakan calon pada setiap siklus operasi yang disebut generasi. Struktur umum pada Algoritma Genetika yaitu:

1. Representasi kromosom.
2. Evaluasi dengan menghitung *fitness*.
3. Proses *crossover* untuk mendapatkan individu baru.
4. Proses mutasi yang untuk meningkatkan variasi dalam populasi.
5. Proses seleksi untuk membentuk populasi baru.

Beberapa hal yang termasuk kelebihan dari Algoritma Genetika adalah sebagai berikut [14]:

1. Mengoptimalkan dengan variabel kontinu atau diskrit,
2. Tidak memerlukan informasi derivatif,
3. Bersamaan pencarian dari sebuah sampling yang luas pada permukaan biaya,
4. Berkaitan dengan sejumlah besar variabel,
5. Baik untuk komputer paralel,
6. Mengoptimalkan permukaan variabel dengan biaya yang sangat kompleks (GA bisa melompat dari minimum lokal),
7. Memberikan daftar variabel yang optimal, bukan hanya solusi tunggal,
8. Dapat menyandikan variabel sehingga optimasi dilakukan dengan mengkodekan variabel, dan
9. Bekerja dengan data numerik yang dihasilkan, data eksperimen, atau analitis fungsi.

Dalam tugas akhir ini, algoritma genetika dipergunakan untuk mengoptimalkan dan membandingkannya dengan hasil dari random forest

BAB III

METODE PENELITIAN

Bab ini membahas mengenai metodologi sistem yang digunakan untuk menyelesaikan Tugas Akhir. Pembahasan metodologi sistem diawali dengan penjelasan tentang objek penelitian, peralatan yang digunakan, dan tahap penelitian. Diagram alur penelitian akan diperlihatkan pada Gambar 3.1.

3.1. Studi Literatur

Pada tahap ini dilakukan pengumpulan teori pendukung mengenai dengan mencari referensi yang menunjang penelitian yang berupa tugas akhir, jurnal, buku, maupun artikel. Dalam tugas akhir ini, studi literatur yang dilakukan mengenai data mining, random forests, dan Algoritma genetika

3.2. Pengumpulan Data

Pengumpulan data merupakan tahap untuk mengumpulkan data yang diperlukan dalam pengerjaan Tugas Akhir ini. Pada Tugas Akhir ini, data diambil secara sekunder berupa data karyawan dan perilaku dalam bekerja. Data sekunder akan di dapat dari perusahaan yang bergerak dibidang pelayaran shipping company, yang salah satu fungsinya sebagai Transportir pemuat Cargo bijih besi maupun batubara. Variabel data yang akan dipakai dalam tugas akhir ini antara lain

Atribut data yang akan dipakai dalam Tugas Akhir ini antara lain:

posisi	:	Menjelaskan posisi pekerjaan yang dimana terdapat 31 jenis posisi, antara lain CHIEF OFFICER TUGBOAT, SUPERVISOR HUMAN RESOURCES, ENGINEER OIL BARGE, dll.
--------	---	--

Pendidikan	:	Latar belakang Pendidikan yang dibagi menjadi 40 jenis
Usia Numerik	:	Usia dari pekerja tersebut
Tanggungan	:	Berisikan informasi tentang jumlah tanggungan (keluarga) pekerja tersebut
Status	:	Status penerimaan hasil wawancara yang memiliki nilai: <ol style="list-style-type: none"> 1 (TP dipanggil = lolos tahap 1) 2 (SP dipanggil = lolos tahap 2)
Keterangan	:	Menjelaskan mengenai keterangan apakah pegawai tersebut tetap bekerja atau apabila pekerja tersebut berhenti maka disertakan juga alasan pengunduran diri, misal naik jabatan atau pengunduran diri dikarenakan PHK
Class	:	Keterangan apakah pekerja tersebut diterima atau ditolak. Ada 2 nilai dalam atribut ini, yaitu : <ol style="list-style-type: none"> 0 (ditolak) 1 (diterima)

3.3. Pengolahan Data

Pengolahan data yang dimaksud adalah mengolah data sekunder yang berupa tabel *excel* menjadi data yang telah menghasilkan pola sebagai tujuan dari Tugas Akhir ini. Untuk menjadikan data sekunder menjadi pola akhir, ada beberapa tahap yang harus dilalui. Tahap-tahap yang harus dilalui ini sesuai dengan KDD yang telah dijelaskan pada Bab 2. Dalam tugas akhir ini, tahap-tahap KDD yang harus dilalui antara lain :

3.3.1. Pre-Processing

Proses ini merupakan proses yang dilakukan untuk membuat data mentah menjadi *training data* (data yang dapat digunakan). Untuk tugas akhir ini, *pre-processing* yang dilakukan ada 2 tahap, yaitu :

Pembersihan Data (*Data Cleaning*) merupakan proses menghapus data *noise* (pengganggu) dan mengisi data yang hilang. Contoh *noise* yang dapat dihilangkan yaitu *missing value data*. *Missing value data* dapat dihilangkan jika tidak ada perubahan atau efek penghilangannya. Data yang dipakai dalam proses ini masih berupa format .excel. Proses ini menghilangkan data dengan atribut yang tidak lengkap.

Transformasi Data merupakan proses pengubahan data ke dalam format tertentu untuk dapat diproses ke dalam *data mining*. Data yang sudah dibersihkan dari *noise*, akan diubah formatnya menjadi format yang dipakai dalam *data mining* agar bisa diproses. Dalam proses ini data akan dibagi menjadi 2 macam, yaitu data set dan data training.

3.3.2. *Data Mining*

Proses ini merupakan proses ekstraksi pola-pola penting dalam basis data yang akan dianalisis. Hal ini dilakukan setelah *pre-processing* selesai. Dalam tugas akhir ini, metode data mining yang dilakukan adalah random forest yang dioptimalkan menggunakan algoritma genetika. *Pre-processing* yang dilakukan pada kedua algoritma sama.

Terdapat beberapa tahap dalam proses ini, yaitu:

Perancangan Algoritma

Dalam tahap ini, penulis akan membangun sebuah tree dari algoritma. Data yang digunakan adalah data training yang telah melalui pre-procesing. Keluaran dari tahap ini adalah rules yang nantinya akan membentuk kelas-kelas berlabel jenis penerimaan tenaga kerja.

Pengujian Algoritma

Dalam tahap ini akan dibagi menjadi data training dan data testing. Data training akan dimasukkan ke dalam *rules* yang nantinya akan menghasilkan pola dalam kelas-kelas *classification*.

Analisis dan Hasil Pola

Tahap ini akan menganalisis pola yang telah terbentuk dan dilakukan beberapa kali pengujian sampai didapat hasil yang paling akurat.

Setelah melakukan tahap-tahap data mining diatas akan diketahui hasil polanya. Jika proses ini berhasil, maka akan didapat angka akuransi yang nantinya dapat dianalisis untuk mencapai tujuan dari Tugas Akhir ini.

3.3.3. Analisis Pola dan Representasi Data

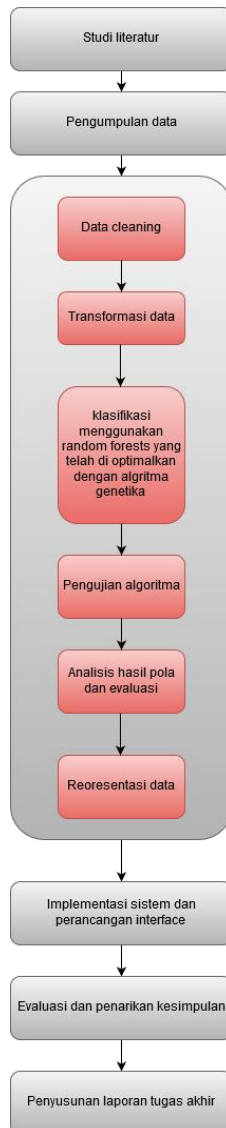
Tahap ini merupakan proses penarikan kesimpulan dari informasi atau pola yang sudah diperoleh dari proses *data mining*.

3.4. Perancangan Interface dan Implementasi Sistem

Pada tahap ini, penulis membuat visualisasi dari hasil *data mining* tersebut agar mudah dipahami oleh pembaca. Pengujian akan dilakukan beberapa kali sampai dipastikan tidak ada eror. Pengujian ini akan mempengaruhi pengambilan keputusan sebagai pertimbangan untuk mengetahui keakuratan metode random forest yang dioptimalkan dengan menggunakan algoritma genetika.

3.5. Evaluasi dan Kesimpulan

Setelah dilakukan analisis dan pembahasan maka dapat ditarik suatu kesimpulan dan saran sebagai masukan untuk pengembangan penelitian lebih lanjut.



Gambar 3.1. Diagram Alur Penelitian

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan analisis data, perancangan gambaran umum sistem, perancangan proses algoritma, perancangan data, dan perancangan antar muka sistem. Sistem ini memiliki input berupa data mentah yang berasal dari database yang selanjutnya dilakukan *pre-processing* untuk menghilangkan noise pada data yang berupa duplikasi data, *missing value data* dan transformasi data. Hasil dari *pre-processing* ini berupa data kategorikal yang siap digunakan sebagai input untuk menjalankan algoritma. Dalam menjalankan algoritma, tidak bisa dilakukan secara bersamaan. Akan tetapi, kedua algoritma dapat dijalankan dengan satu kali tahap *pre-processing*. Hasil dari proses ini adalah *rules* pada tiap algoritma, yang selanjutnya akan di testing sehingga memunculkan angka keakurasian pada tiap algoritma tersebut. dijelaskan secara detail tentang spesifikasi kebutuhan perangkat lunak yang berupa informasi mengenai fungsi yang dibutuhkan untuk performansi dan *interface*.

4.1 Gambaran Umum Perangkat Lunak

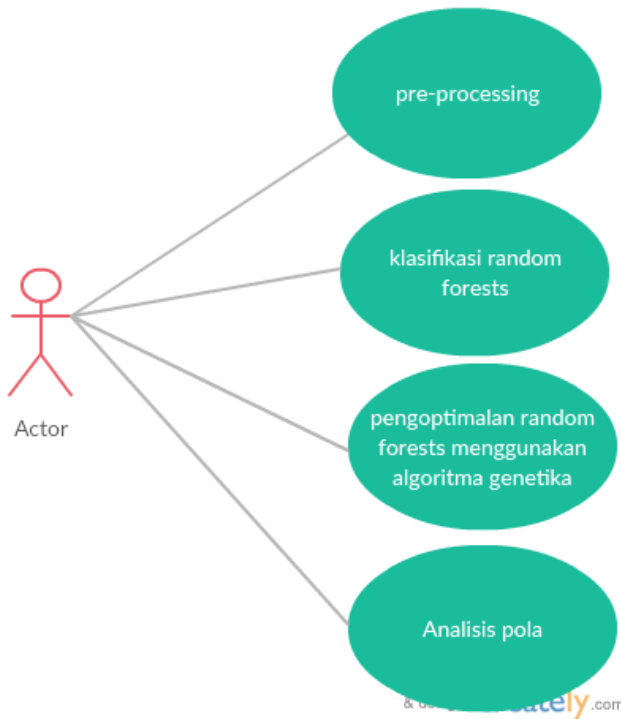
Secara keseluruhan program ini dibuat dengan menggunakan bahasa pemrograman Java. Adapun Java *library* atau perpustakaan yang digunakan adalah WEKA. WEKA adalah *library* yang memudahkan dalam penggunaan *data mining*, khususnya *random forests* oleh para pemrogram untuk membangun aplikasinya.

Perangkat lunak ini berfungsi sebagai aplikasi yang memudahkan dalam perhitungan akurasi dari suatu masalah menggunakan *random forests* yang dioptimalkan dengan algoritma genetika sehingga *decision maker* dapat mengambil keputusan dengan hasil keakuratan yang telah dihasilkan.

Perangkat lunak yang akan dibangun dapat membantu pihak-pihak dalam menganalisis data dalam keadaan kecil maupun besar. Klasifikasi ini dapat memprediksi pola dari data histori.

4.2 Usecase Diagram

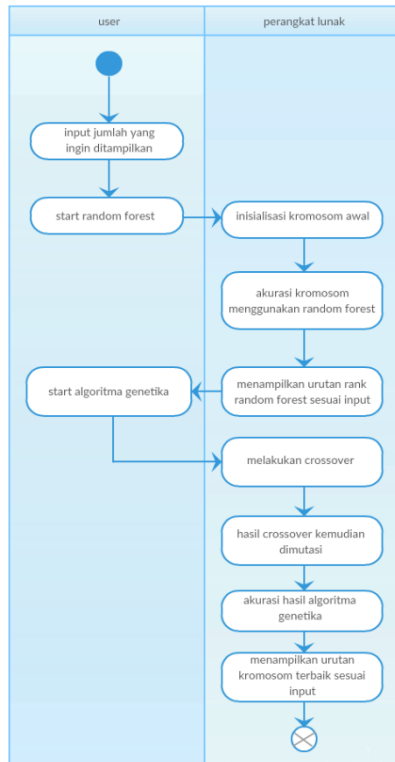
Use case diagram menggambarkan interaksi apa saja yang dapat dilakukan di dalam sistem dan siapa saja aktor yang akan menjalankan interaksi tersebut. *Use case diagram* tersebut dapat dilihat pada Gambar 4.1.



Gambar 4.1 *Usecase Diagram* Sistem Aplikasi

4.3 Activity Diagram

Berdasarkan *use case diagram* yang ditunjukkan pada Gambar 4.1, kemudian dibuat suatu *activity diagram* untuk menggambarkan alur proses dari setiap interaksi.



Gambar 4.2 *Activity Diagram* Sistem Algoritma Genetika untuk optimalisasi data dalam proses klasifikasi penerimaan tenaga kerja baru menggunakan metode random forest.

4.4 Pendefinisian Individu

Individu menyatakan satu solusi yang mungkin. Individu sendiri dapat dikatakan sebagai kromosom, yang merupakan kumpulan gen. Gen ini dapat berisi biner, *integer*, *float*, *double*, karakter, atau bahkan kombinatorial. Dalam penelitian kali ini, gen akan berisi nilai biner, yaitu nilai biner dari nilai pada setiap fitur. Fitur yang digunakan berjumlah 6 yaitu, posisi, pendidikan, usia numerik, tanggungan, status, dan keterangan. Satu individu terdiri dari 6 kromosom dan individu tersebut merupakan solusi dari permasalahan yang ada dalam penelitian kali ini.

4.5 Inisialisasi Kromosom

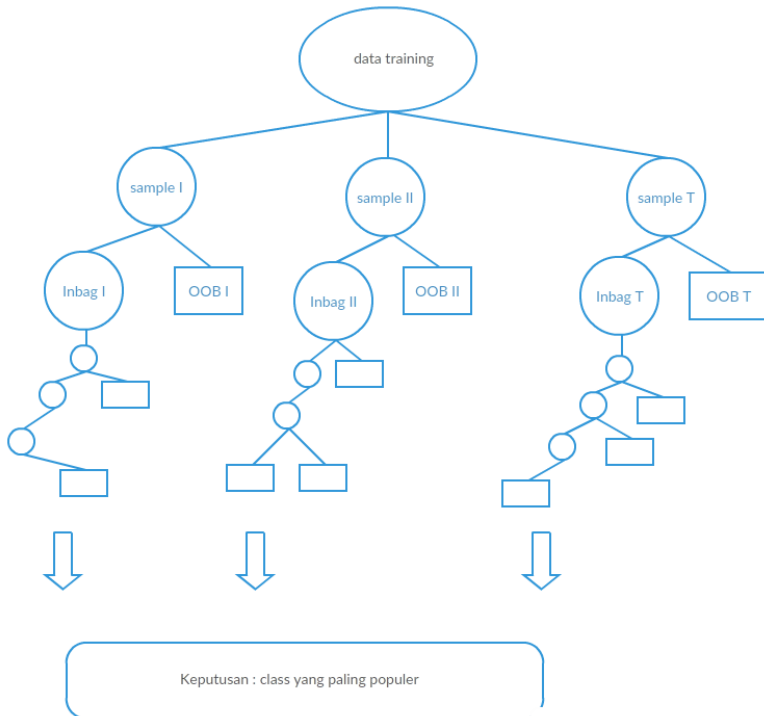
Inisialisasi adalah tahap dimana Populasi Awal akan ditentukan. Karena kromosom bisa direpresentasikan dengan binary string dan kita memiliki 64 fitur (Ada 6 fitur dalam data sehingga banyak kemungkinan kromosom yang terjadi adalah $2^6 = 64$) maka untuk membuat inisialisasi awal kromosom kita dapat merandom angka dari 1-63 karena 0 akan menghilangkan semua fitur.

4.6 Random Forest

Pada weka random forest memiliki parameter yaitu -U untuk unpruned tree. Inisialisasi RandomForest dengan new RandomForest. Kemudian build classifier (process learning) menggunakan data training. Untuk uji coba memakai Evaluation dengan input tree hasil training random forest serta data set training, kemudian menghitung nilai benar /jumlah data test untuk mendapatkan nilai akurasi.

Gambar 4.3 menampilkan alur kerja random forests. Dalam pembentukan tree, random forest akan melakukan training pada sampel data. Pengambilan sample dengan cara sampling with replacement. Sebanyak sepertiga dari sample akan digunakan

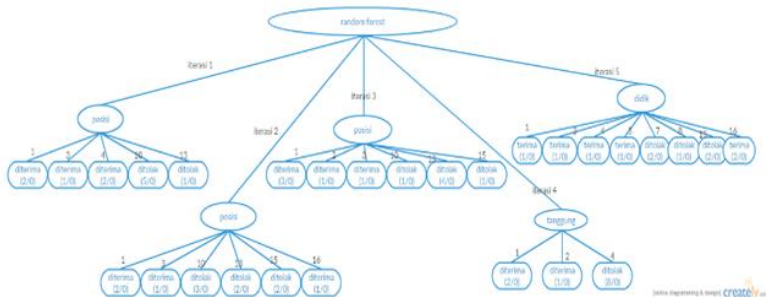
untuk menentukan data out of bag (OOB). Penentuan data out of bag (OOB) dilakukan untuk mengestimasi eror dan menentukan variable importance. Setelah seluruh tree terbentuk, maka proses klasifikasi akan berjalan.



Gambar 4.3 Alur kerja random forest

Setelah klasifikasi berjalan, maka akan dilanjutkan dengan inisialisasi. Inisialisasi dilakukan sebanyak jumlah yang diinginkan oleh user dan akan di urutkan berdasarkan nilai akurasi tertinggi ke rendah.

Struktur random forest yang diperlihatkan pada Gambar 4.4 adalah contoh 11 data acak dari keseluruhan total data yang berjumlah 126 dengan bagging sebanyak 5 iterasi dan persentase split sebanyak 70%.



Gambar 4.4 Struktur Random Forest

Pada Gambar 4.4 terlihat untuk bagging iterasi 1 menghasilkan pohon “posisi” dengan ketentuan:

- Posisi 1: diterima sebanyak 2
- Posisi 3: diterima sebanyak 1
- Posisi 4: diterima sebanyak 2
- Posisi 10: ditolak sebanyak 5
- Posisi 13: ditolak sebanyak 1

Untuk bagging iterasi 2 menghasilkan pohon “posisi” dengan ketentuan:

- Posisi 1: diterima sebanyak 2
- Posisi 3: diterima sebanyak 1
- Posisi 10: ditolak sebanyak 3
- Posisi 13: ditolak sebanyak 2

- Posisi 15: ditolak sebanyak 2
- Posisi 16: diterima sebanyak 1

Untuk bagging iterasi 3 menghasilkan pohon “posisi” dengan ketentuan:

- Posisi 1: diterima sebanyak 3
- Posisi 2: diterima sebanyak 1
- Posisi 3: diterima sebanyak 1
- Posisi 10: ditolak sebanyak 1
- Posisi 13: ditolak sebanyak 4
- Posisi 15: ditolak sebanyak 1

Untuk bagging iterasi 4 menghasilkan pohon “tanggung” dengan ketentuan:

- Tanggung 1: diterima sebanyak 2
- Tanggung 2: diterima sebanyak 1
- Tanggung 4: ditolak sebanyak 8

Dan untuk bagging iterasi 5 menghasilkan pohon “didik” dengan ketentuan:

- Didik 1: diterima sebanyak 1
- Didik 3: diterima sebanyak 1
- Didik 4: diterima sebanyak 1
- Didik 5: diterima sebanyak 1
- Didik 7: ditolak sebanyak 2
- Didik 8: ditolak sebanyak 1
- Didik 15: ditolak sebanyak 2
- Didik 16: diterima sebanyak 2

Berikut ini adalah contoh data awal (Tabel 4.1) dan data hasil resample (Tabel 4.2) sebelum diolah dengan teknik random forest:

Tabel 4.1. Data awal

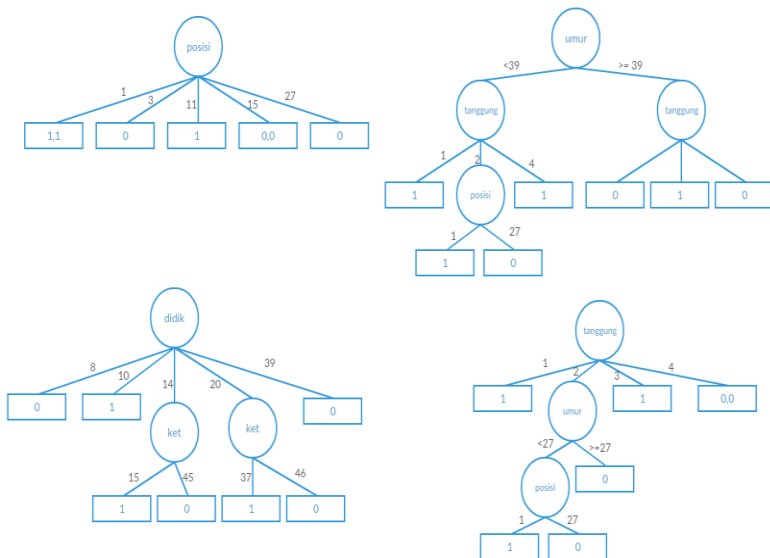
Data ke	X1...	X6	Y
1	2	1	1
2	3	2	1
3	1	3	0
4	4	21	0
5	16	45	0
.			
.			
.			
N	10	47	1

Tabel 4.2. Data resample

Data ke	X1	X3	X4	Y
1	3	34	4	0
2	27	27	2	0
3	1	40	4	0
4	1	27	2	1
5	1	38	1	1
6	15	41	2	0
7	11	42	3	1

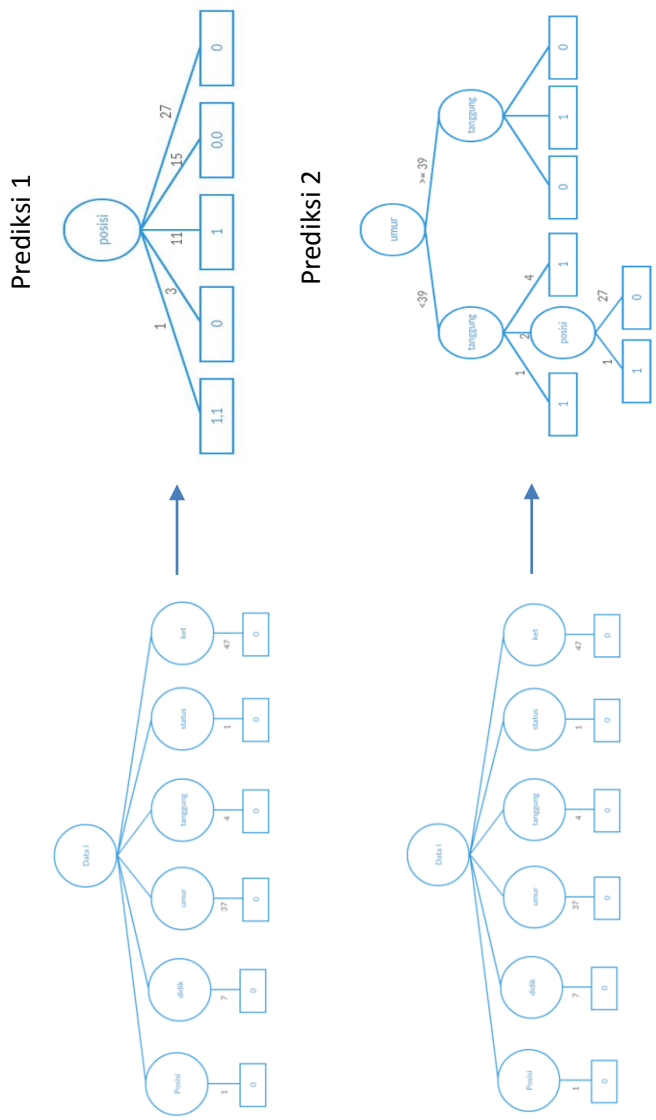
Inti dari algoritma random forest adalah bagaimana mengambil data resample dan variable faktor secara acak. Pada contoh ini digunakan variabel faktor sebanyak 3 seperti yang terlihat pada Tabel 4.2 yaitu X1, X3, DAN X4.

Dari pengambilan data acak tersebut, maka akan terbentuk kumpulan tree yang nantinya akan membentuk random forest. Gambar 4.5 adalah struktur dari random forest yang telah terbentuk dari pengambilan data acak.



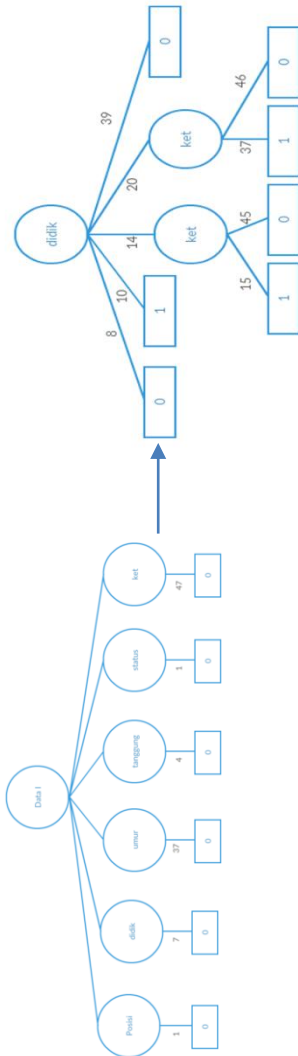
Gambar 4.5 Struktur Random Forest

Setelah terbentuk struktur *Random Forest*-nya, kita bisa memperhentikan nilai akurasi dengan menggunakan data testing. Gambar 4.6 dan Gambar 4.7 adalah contoh perhitungan akurasi menggunakan data ke-1.

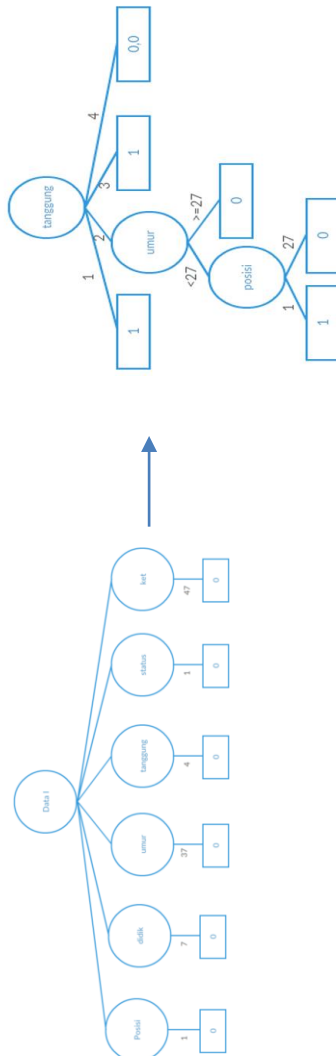


Gambar 4.6 Prediksi Random Forest

Prediksi 3



Prediksi 4



Gambar 4.7 Prediksi Random Forest

Tabel 4.3. Hasil Prediksi

Dat a ke	X1. ..	X 6	Y	Predi ksi 1	Predi ksi 2	Predi ksi 3	Predi ksi 4	Predi ksi akhir
1	1	4 7	0	1	1	-	0	1
2	15	4 8	0	0	-	-	-	0
3	2	4 7	0	-	1	0	0	0

Tabel 4.3 diambil dengan aturan mayoritas semua pohon keputusan. Nilai akurasi didapatkan dari jumlah total prediksi benar dibagi dengan total keseluruhan data yaitu $2/3$ dan menghasilkan akurasi sebesar 66,7%

.Source Codenya:

```
public int compareTo(chromosomeData o) {
    return (int) (o.accuracy - this.accuracy);
}
```

```

public static double doRandomForest(Instances training,
Instances testing) throws Exception
{
    double accuracy;

    String[] options = new String[1];
    options[0] = "-U";
    RandomForest tree = new RandomForest();
    tree.setOptions(options);
    tree.buildClassifier(training);
    Evaluation eval = new Evaluation(testing);
    eval.evaluateModel(tree, testing);
    accuracy = (eval.correct()/56)*100;

    return accuracy;
}

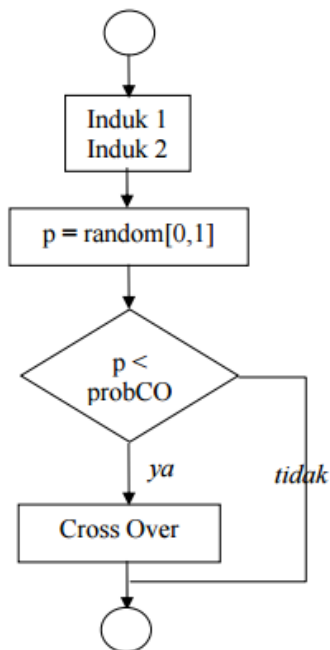
```

4.7 Crossover pada Algoritma Genetika

Pada algoritma genetika, kromosom hasil inisialisasi akan diproses sesuai tahapan algoritma genetika yaitu crossover dan mutasi. Proses algoritma genetika menggunakan perulangan sebanyak 30 kali. Setiap perulangan akan dipilih 2 kromosom secara random dengan kriteria akurasi rank 10 besar.

Setelah terpilih kromosom 1 dan 2, maka akan di *crossover* dengan tipe *one point crossover*. Pada fungsi ini apabila iterasi

kurang dari point maka nilai kromosom sama dengan nilai kromosom awal tetapi jika melebihi nilai point maka nilainya akan diganti dengan kromosom lainnya.



Gambar 4.8 Alur kerja *crossover*

Source Codenya:

```
public static boolean[] crossover1(boolean bits1[],boolean
bits2[],int point)
{
    boolean[] bits = new boolean[6];
    int i;
    for(i=0;i<6;i++)
    {
        if(i<point){
            bits[i]=bits1[i];
        }
        else
        {
            bits[i]=bits2[i];
        }
    }
}
```

4.8 Mutasi pada Algoritma Genetika

Setelah selesai crossover maka setiap kromosom akan dilakukan mutasi dengan random rate mutasi. Pada tahap ini setiap bit dalam kromosom akan dicek satu persatu dengan suatu nilai random 0 dan 1. apabila nilai random bernilai 0 maka nilai kromosom index tsb di negasikan ($0 \rightarrow 1, 1 \rightarrow 0$) apabila 1 maka tetap.

Source Codenya:

```
public static boolean[] mutation(boolean input[])
{
    boolean[] bits = new boolean[6];

    int max =2;
    int min= 0;
    int i;
    Random r = new Random();
    int randInt;

    for(i=0;i<6;i++)
    {
        randInt = r.nextInt(max-min) + min;

        if(randInt==0){
            if(input[i]==false)
            {
                bits[i]=true;
            }
            else
            {
                bits[i]=false;
            }
        }
    }

    else
```

```

{
    bits[i] = input[i];

}

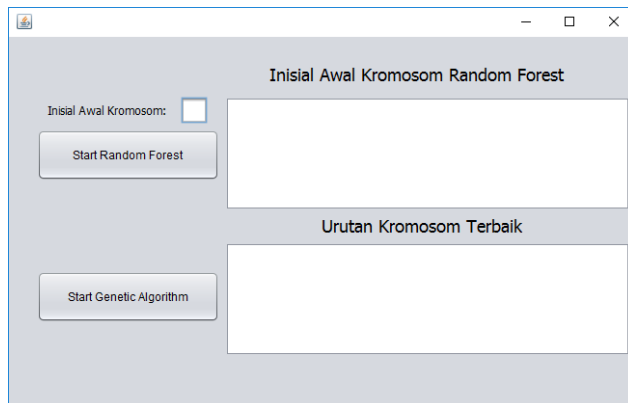
}

return bits;
}

```

4.9 Desain *Interface*

Subbab ini akan membahas mengenai interface awal program yang telah dibuat dengan menggunakan bahasa pemrograman Java. Gambar 4.9 merupakan desain tampilan awal yang muncul pada saat program dijalankan.



Gambar 4.9 Form utama

Setelah form utama, user dapat melihat beberapa menu yang terdapat pada pilihan menu bar di dalam form utama. Masing-masing menu bar tersebut memiliki menu item dan fungsi yang berbeda-beda.

Pertama user akan diminta untuk menginputkan jumlah berapa kromosom yang ingin ditampilkan. Kemudian saat user mengklik tombol start random forest, maka akan muncul hasil dari random forest. Untuk melihat hasil dari algoritma genetika, maka user harus menekan tombol start genetic algorithm dan akan muncul urutan kromosom terbaik sesuai input yang telah dimasukkan oleh user.

BAB V

PENGUJIAN DAN PEMBAHASAN

Bab ini akan menampilkan hasil training berupa *rules*, hasil testing berupa angka keakuratan, dan selanjutnya angka tersebut akan dibandingkan. Hasil pengujian inilah yang akan digunakan dalam perumusan kesimpulan dan saran.

5.1 Pre Processing dan Pembagian Data Uji

Pre Processing dilakukan diawal sebelum menjalankan perangkat lunak ini. Data asli berupa string/kalimat yang kemudian dikonversi menjadi id berupa integer agar mudah untuk diproses. Data seperti nomor telepon, nomor pelamar, alamat, tanggal terima cv, NIK, dan nama dihapus. Kemudian penambahan atribut umur yang didapatkan dari tanggal lahir. Data lengkap sebelum dan sesudah pre-processing akan dicantumkan pada lembar lampiran. Hasil dari tahap *pre processing* ini terlihat pada Gambar 5.1 dan Gambar 5.2 dibawah ini.

K	L	M	N	O	P	Q	R	S	T
Posisi yang Didamar	Pendidikan	Tanggal Lahir	Tanggal CV Diterima	Status	NIK		name	Tanggal lahir	Tanggungan
2ND OFFICER TUGBOAT	Diploma 3 Nautika	13/06/1974	4/02/2013	TP DIPANGGIL	0011302864		AMIRUDDIN	13 Juni 1974	K2
2ND ENGINEER TUG BOAT	CERTIFICATE ATT-IV	25/12/1990	6/03/2013	TP DIPANGGIL	0011303570		III. ROSULLE U.S SAPUTRA	25 Desember 1990	TK0
CHIEF OFFICER TUGBOAT	CERTIFICATE BASIC SAFETY TRAINING	10/01/1980	11/03/2013	TP DIPANGGIL	0011303572		AWALUDDIN	10 Januari 1980	K1
2ND OFFICER TUGBOAT	BUKU PELAUT	25/01/1981	18/03/2013	SP DIPANGGIL	0011112219		BLANJET RIYADI	10 November 1972	K2

Gambar 5.1 Hasil Tampilan Data sebelum *pre-processing*

	A	B	C	D	E	F	G
1	posisi	didik	usia	tanggung	stat	ket	class
2	1	1	39		1	1	1
3	2	2	22		2	1	1
4	3	3	33		3	1	1
5	1	4	32		1	2	4
6	4	5	29		3	1	5
7	5	5	41		1	1	6
8	6	2	41		3	1	7

Gambar 5.2 Hasil *Pre Processing*

Uji coba pada perangkat lunak dalam penelitian Tugas Akhir ini dilakukan dengan membagi dataset menjadi 2, yaitu data training dan data testing dengan perbandingan 2:1. Data training digunakan oleh algoritma klassifikasi untuk membentuk sebuah model classifier. Model ini merupakan representasi pengetahuan yang akan digunakan untuk prediksi kelas data baru yang belum pernah ada. Data testing digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar. Kemudian dataset akan dikonversi menjadi bentuk arff.

5.2 Representasi Data

Dalam algoritma genetika fitur dapat direpresentasikan sebagai chromosome. Apabila ada 6 fitur maka akan ada 6 kromosom dengan kromosom 1 merupakan representasi fitur 1 dst. Apabila kromosom bernilai True maka fitur yang direpresentasikan kromosom dipakai untuk train dan testing.

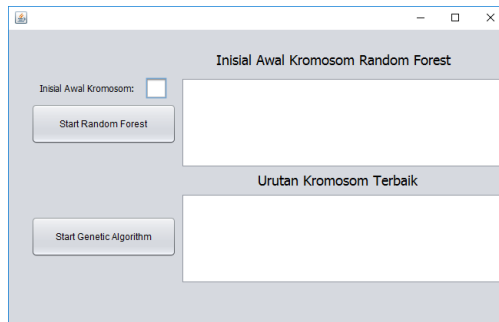
	A	B	C	D	E	F	G
1	posisi	didik	usia	tanggung	stat	ket	class
2	1	1	39		1	1	1
3	2	2	22		2	1	1
4	3	3	33		3	1	3
5	1	4	32		1	2	4
6	4	5	29		3	1	5
7	5	5	41		1	1	6
8	6	2	41		3	1	7

Gambar 5.3 Hasil *Pre Processing*

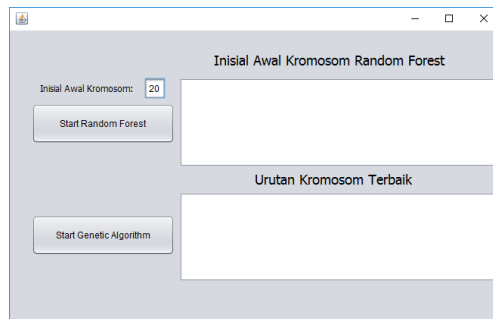
Apabila kromosom bernilai (T,F,T,T,F,T) dengan data seperti Gambar 5.3, maka fitur yang digunakan adalah posisi, usia, tanggung, dan ket. Ada 6 fitur dalam data sehingga banyak kemungkinan kromosom yang terjadi adalah $2^6 = 64$.

5.3 Inisialisasi Kromosom

Karena kromosom bisa direpresentasikan dengan binary string dan kita memiliki 64 fitur maka untuk membuat inisialisasi awal kromosom kita dapat merandom angka dari 1-63 karena 0 akan menghilangkan semua fitur. Jumlah inisialisasi kromosom nanti akan diinputkan oleh user secara manual.



Gambar 5.4 Tampilan awal sebelum input jumlah inisialisasi



Gambar 5.5 Input jumlah inisialisasi

Pada pengujian kali ini, akan di inputkan jumlah inisialisasi awal kromosom sebanyak 20.

Input merupakan nilai random integer dari 1-63. Apabila hasil random menunjukkan angka 3 maka representasi kromosomnya: 0,0,0,0,1,1 sehingga fitur yang digunakan nomer 5 dan 6 (karena bernilai 1 atau true).

5.4 Akurasi Random Forest



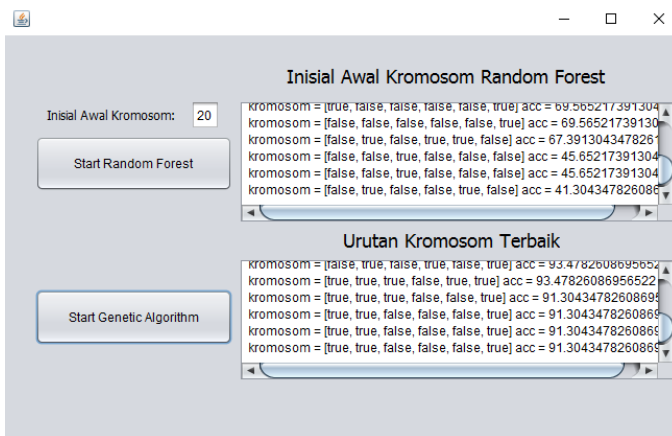
Gambar 5.6 hasil akurasi random forest

Saat user memasukkan/menginput jumlah inisialisasi kromosom awal sesuai Gambar 5.6 maka akan ditampilkan jumlah kromosom sesuai input beserta nilai akurasinya. Berdasarkan *running program* tersebut diperoleh hasil sebagai berikut:

1. kromosom = [true, false, false, true, false, true] acc = 95.65217391304348
2. kromosom = [false, true, true, true, false, true] acc = 93.47826086956522
3. kromosom = [true, true, false, true, false, true] acc = 93.47826086956522
4. kromosom = [true, true, true, false, false, true] acc = 91.30434782608695
5. kromosom = [true, true, false, false, false, true] acc = 91.30434782608695
6. kromosom = [false, true, true, false, true, true] acc = 84.78260869565217
7. kromosom = [false, false, false, true, true, false] acc = 84.78260869565217
8. kromosom = [false, false, true, false, true, true] acc = 82.6086956521739
9. kromosom = [true, true, true, true, false, false] acc = 78.26086956521739
10. kromosom = [false, false, true, true, true, false] acc = 78.26086956521739
11. kromosom = [false, true, true, true, true, false] acc = 73.91304347826086
12. kromosom = [false, false, false, false, true, true] acc = 69.56521739130434
13. kromosom = [true, true, true, false, false, false] acc = 69.56521739130434
14. kromosom = [false, true, false, false, false, true] acc = 69.56521739130434
15. kromosom = [true, false, false, false, false, true] acc = 69.56521739130434
16. kromosom = [false, false, false, false, false, true] acc = 69.56521739130434
17. kromosom = [false, true, false, true, true, false] acc = 67.3913043478261

18. kromosom = [false, false, true, false, false, false] acc = 45.65217391304348
19. kromosom = [false, false, true, false, false, false] acc = 45.65217391304348
20. kromosom = [false, true, false, false, true, false] acc = 41.30434782608695

5.5 Akurasi Algoritma Genetika



Gambar 5.7 Hasil akurasi setelah di optimalisasi dengan algoritma genetika

Dalam tahapan ini kromosom hasil inialisasi akan diproses sesuai tahapan GA yaitu crossover dan mutasi. Proses algoritma genetika menggunakan 30 kali perulangan. Setiap perulangan akan dipilih 2 kromosom secara random dengan kriteria akurasi rank 10 besar. Setelah terpilih 2 chromosome 1 dan 2, maka akan dilakukan crossover dengan tipe one point crossover.

Pada fungsi ini apabila iterasi kurang dari point maka nilai kromosom sama dengan nilai kromosom awal tetapi jika melebihi nilai point maka nilainya akan diganti dengan kromosom lainnya.

Setelah selesai crossover maka setiap kromosom akan dilakukan mutasi dengan random rate mutasi. Pada tahap ini setiap bit dalam kromosom akan dicek satu persatu dengan suatu nilai random 0 dan 1. apabila nilai random bernilai 0 maka nilai kromosom index tsb di negasikan ($0 \rightarrow 1, 1 \rightarrow 0$) apabila 1 maka tetap.

Berdasarkan *running program* tersebut diperoleh hasil sebagai berikut:

1. kromosom = [true, false, false, true, false, true] acc = 95.65217391304348
2. kromosom = [true, false, false, true, false, true] acc = 95.65217391304348
3. kromosom = [true, false, true, true, true, true] acc = 95.65217391304348
4. kromosom = [true, false, true, true, true, true] acc = 95.65217391304348
5. kromosom = [true, false, true, true, true, true] acc = 95.65217391304348
6. kromosom = [true, false, false, true, true, true] acc = 95.65217391304348
7. kromosom = [true, true, true, true, false, true] acc = 95.65217391304348
8. kromosom = [false, true, true, true, false, true] acc = 93.47826086956522
9. kromosom = [true, true, false, true, false, true] acc = 93.47826086956522
10. kromosom = [true, true, true, false, true, true] acc = 93.47826086956522
11. kromosom = [true, false, true, true, false, true] acc = 93.47826086956522
12. kromosom = [true, true, false, true, true, true] acc = 93.47826086956522
13. kromosom = [false, true, false, true, false, true] acc = 93.47826086956522

14. kromosom = [true, false, true, true, false, true] acc = 93.47826086956522
15. kromosom = [false, true, false, true, false, true] acc = 93.47826086956522
16. kromosom = [true, true, true, false, true, true] acc = 93.47826086956522
17. kromosom = [true, true, true, false, false, true] acc = 91.30434782608695
18. kromosom = [true, true, false, false, false, true] acc = 91.30434782608695
19. kromosom = [true, true, false, false, false, true] acc = 91.30434782608695
20. kromosom = [true, true, false, false, false, true] acc = 91.30434782608695

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi tentang beberapa kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan. Di samping itu, pada bab ini juga dimasukkan beberapa saran yang dapat digunakan jika penelitian ini ingin dikembangkan.

6.1 Kesimpulan

Berdasarkan analisis terhadap hasil pengujian yang telah dilakukan terhadap klasifikasi random forest dan klasifikasi random forest yang dioptimalkan dengan algoritma genetika dengan 126 data tenaga kerja, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Penulis dapat melakukan klasifikasi random forest dengan menggunakan perangkat lunak yang dibangun dengan bahasa pemrograman java, sehingga didapatkan pola penerimaan tenaga kerja beserta angka keakuratan.
2. Hasil keakuratan pola yang dihasilkan oleh random forest yang dioptimalkan dengan algoritma genetika lebih tinggi dibanding dengan hanya menggunakan random forest saja. Dibuktikan pada hasil perhitungan akurasi random forest yang berkisar antara 40%-95%, sedangkan hasil perhitungan akurasi dengan pengoptimalan algoritma genetika menunjukkan hasil berkisar antara 91%-95%.
3. Didapatkan kromosom-kromosom terbaik hasil dari klasifikasi random forest yang dioptimalkan dengan algoritma genetika sebagai acuan untuk menerima tenaga kerja baru.

6.2 Saran

Berdasarkan hasil yang dicapai pada Tugas Akhir ini, ada beberapa hal yang penulis sarankan untuk pengembangan selanjutnya yaitu:

1. Data yang dipakai dalam perangkat lunak ini masih sedikit dan bervariasi. Dalam perhitungan data mining, sebaiknya data harus dalam jumlah yang banyak karena mempengaruhi pola dan hasil keakuratan yang didapatkan. Untuk penelitian selanjutnya dapat menggunakan data yang lebih banyak.
2. Pengembangan perangkat lunak ini dapat dilakukan dengan menambah menu input data baru demi manfaat yang lebih baik.
3. Berdasarkan pola yang dihasilkan maka diperlukan metode terkait pengambilan keputusan untuk penerimaan tenaga kerja selanjutnya

DAFTAR PUSTAKA

- [1] Gorunescu, Florin. (2011). **“Data Mining: Concepts, Models, and Techniques”**. Verlag Berlin Heidelberg. Springer
- [2] Jantan. Hamidah, dkk. (2009). **“Knowledge Discovery Techniques for Talent Forecasting in Human Resource Application”**. International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering Vol:3, No:2,
- [3] Han, J, Micheline K, Jian Pei. (2012). **“Data Mining Concept and Techniques 3rd Edition”**. Morgan Kaufman Publisher. USA
- [4] Elyan, Eyad. dan Mohamed Medhat Gaber. (2016). **“A Genetic Algorithm Approach to Optimising Random Forests Applied to Class Engineered Data”**. School of Computing Science and Digital Media, Robert Gordon University, UK.
- [5] Chien, C.F. dan L.F. Chen. (2008). **“Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry”**. Expert Systems and Applications. (pp 280-290).
- [6] Chen, K.K., dkk (2007). **“Constructing a Web-based Employee Training Expert System with Data mining Approach”**. The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-services (CEC-EEE 2007).
- [7] Adi Saputro, Halim, Wayan Firdaus Mahmudy dan Candra Dewi. (2015). **“Implementasi Algoritma Genetika untuk Optimasi Penggunaan Lahan Pertanian”**. Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

- [8] Mahmudy, Wayan Firdaus. (2013). “**Algoritma Evolusi**”. Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya. Malang
- [9] Kadir, A. (2005). **Dasar Pemrograman Java 2**. Yogyakarta: ANDI.
- [10] Hariyanto, B. (2011). **Esensi-Esensi Bahasa Pemrograman Java**. Bandung: Informatika.
- [11] Lievens, Van Dam, & Anderson (2002). **Recent Trends and Challenges in Personnel Selection**. ABI/INFORM Global pg. 580.
- [12] Berry, M.J.A dan Linoff, G. (1997). **Data Mining Techniques: For Marketing, Sales, and Customer Support**. John Wiley & Sons.
- [13] Breiman, L. (2001). **Machine Learning**, 45: 5-32.
- [14] Haupt, Randy L., dan Haupt, Sue Ellen. (2004) **Practical Genetic Algorithms Second Edition**. S.1 : John Wiley & Sons.

LAMPIRAN A

Data Karyawan Diterima

No	No Pelamar	Posisi yang Dilamar	Pendidikan	Tanggal Lahir	Tanggal CV Diterima	Status
3	6B13020003	2ND OFFICER TUGBOAT	Diploma 3 Nautika	13/06/1974	4/02/2013	TP DIPANGGIL
17	6B13030024	2ND ENGINEER TUG BOAT	CERTIFICATE ATT-IV	25/12/1990	6/03/2013	TP DIPANGGIL
18	6B13030025	CHIEF OFFICER TUGBOAT	CERTIFICATE BASIC SAFETY TRAINING	10/01/1980	11/03/2013	TP DIPANGGIL
19	6B13030029	2ND OFFICER TUGBOAT	BUKU PELAUT	25/01/1981	18/03/2013	SP DIPANGGIL
24	6B13040035	SUPERVISOR HUMAN RESOURCES	Strata 1 Akuntansi	4/10/1984	11/04/2013	TP DIPANGGIL
26	6B13040039	SUPERINTENDENT FINANCE	Strata 1 Akuntansi	23/12/1971	20/04/2013	TP DIPANGGIL
28	6B13040041	ENGINEER OIL BARGE	CERTIFICATE ATT-IV	10/04/1972	25/04/2013	TP DIPANGGIL
30	6B13050043	A/B OIL BARGE	CERTIFICATE SURVIVAL CRAFT AND RESCUE BOATS	27/07/1977	24/04/2013	TP DIPANGGIL
31	6B13050045	COOK LCT	SMU Lain-lain	28/02/1992	9/05/2013	TP DIPANGGIL

32	6B13050046	CHIEF OFFICER TUGBOAT	CERTIFICATE SURVIVAL CRAFT AND RESCUE BOATS	7/04/1981	13/05/2013	TP DIPANGGIL
33	6B13050047	TECHNICIAN SERVICE ENGINE AND REPAIR	SMK Otomotif	6/04/1992	16/05/2013	TP DIPANGGIL
38	6B13060054	SUPERVISOR CAPTAIN TUG BOAT	KONTRAK 1 (3 BULAN)	25/12/1973	11/06/2013	SP DIPANGGIL
40	6B13060056	CHIEF ENGINEER TUG BOAT	SMU Lain-lain	30/06/1979	10/06/2013	TP DIPANGGIL
41	6B13060058	2ND OFFICER LCT	SMU Lain-lain	11/04/1978	17/06/2013	SP DIPANGGIL
45	6B13070062	ADMIN PORT CLEARANCE	SLTP dan sederajat	23/04/1973	17/05/2013	TP DIPANGGIL
47	6B13080064	2ND OFFICER TUGBOAT	SMK Teknika	3/12/1978	12/08/2013	TP DIPANGGIL
48	6B13080065	2ND ENGINEER TUG BOAT	SMK Perkapalan	12/11/1968	13/08/2013	TP DIPANGGIL
49	6B13080066	CHIEF OFFICER TUGBOAT	SMK Nautika	31/05/1978	13/08/2013	TP DIPANGGIL
50	6B13080068	SPEEDBOAT MOTORIST	SMU Lain-lain	16/11/1989	16/07/2013	TP DIPANGGIL
54	6B13080072	ADMIN PORT CLEARANCE	SMK Mesin	23/06/2013	1/08/2013	TP DIPANGGIL
56	6B13080074	CHIEF OFFICER TUGBOAT	KONTRAK 1 (3 BULAN)	26/10/1975	29/08/2013	TP DIPANGGIL
57	6B13080075	3RD ENGINEER TUGBOAT	SMU IPS	13/08/2013	27/08/2013	TP DIPANGGIL

58	6B13080076	ADMIN GENERAL ADMINISTRATION & SYSTEM	SMK Lan-lain	1/11/1992	28/08/2013	TP DIPANGGIL
59	6B13080079	2ND OFFICER TUGBOAT	SMU IPS	25/06/1976	9/09/2013	TP DIPANGGIL
60	6B13080080	SPEEDBOAT MOTORIST	SMK Otomotif		2/09/2013	TP DIPANGGIL
64	6B13090090	2ND OFFICER TUGBOAT	SMU IPS	22/11/1982	5/10/2013	TP DIPANGGIL
68	6B13090098	2ND ENGINEER TUG BOAT	SMK Teknika	14/04/1990	3/10/2013	TP DIPANGGIL
70	6B13110103	SUPERVISOR CAPTAIN TUG BOAT	SMK Nautika	7/05/1971	1/11/2013	SP DIPANGGIL
74	6B13110108	ADMIN BANK AND CASH	Strata 1 Akuntansi	7/12/1991	7/11/2013	SP DIPANGGIL
75	6B13110110	ADMIN PERSONNEL	SMK Akuntansi	9/12/1991	11/11/2013	SP DIPANGGIL
76	6B13110111	CHIEF OFFICER TUGBOAT	PASSPORT	15/07/1972	18/11/2013	SP DIPANGGIL
77	6B13120113	ADMIN PORT CLEARANCE	SMU IPS	12/07/1988	28/11/2013	SP DIPANGGIL
78	6B13120114	2ND OFFICER TUGBOAT	SLTP dan sederajat	24/04/1979	5/12/2013	SP DIPANGGIL
79	6B13120115	ENGINEER OIL BARGE	SMU IPS	1/02/1979	5/12/2013	SP DIPANGGIL
80	6B13120116	2ND ENGINEER TUG BOAT	SMK Elektronika	5/06/1974	6/12/2013	SP DIPANGGIL
81	6B13120117	CHIEF ENGINEER TUG BOAT	SMK Pelayaran	20/06/1984	9/12/2013	TP DIPANGGIL

84	6B13120120	2ND ENGINEER TUG BOAT	SMK Otomotif	21/10/1987	20/12/2013	TP DIPANGGIL
85	6B14010121	TECHNICIAN ELECTRICAL AND NAVIGATION	SMU IPS	7/12/1991	27/12/2013	TP DIPANGGIL
86	6B14010123	ASSISTANT ADMINISTRATION	SMU Lain-lain	21/09/1989	10/11/2013	TP DIPANGGIL
87	6B14010125	SUPERVISOR CAPTAIN TUG BOAT	SMK Nautika	14/11/1971	3/01/2014	TP DIPANGGIL
90	6B14010128	CHIEF ENGINEER TUG BOAT	KONTRAK 1 (3 BULAN)	8/08/1976	4/01/2014	TP DIPANGGIL
91	6B14010129	2ND ENGINEER TUG BOAT	SMK Perkapalan	31/07/1982	11/01/2014	TP DIPANGGIL
92	6B14010130	2ND ENGINEER TUG BOAT	SMK Otomotif	10/08/1985	7/01/2014	TP DIPANGGIL
94	6B14010136	3RD ENGINEER TUGBOAT	SMK Otomotif	9/06/1987	9/01/2014	TP DIPANGGIL
95	6B14010137	CHIEF OFFICER TUGBOAT	SMK Pelayaran	15/03/1987	8/01/2014	TP DIPANGGIL
96	6B14010141	2ND ENGINEER TUG BOAT	SMK Teknika	1/08/1981	21/01/2014	TP DIPANGGIL
97	6B14010142	2ND ENGINEER TUG BOAT	SMU Lain-lain	15/09/1968	28/01/2014	TP DIPANGGIL
98	6B14020144	CHIEF OFFICER TUGBOAT	SMK Nautika	5/05/1990	4/02/2014	TP DIPANGGIL
99	6B14020145	2ND ENGINEER TUG BOAT	KTP	13/09/1978	4/02/2014	TP DIPANGGIL
102	6B14020149	ADMIN GENERAL ADMINISTRATION & SYSTEM	Strata 1 Teknik Komputer	21/08/1989	8/02/2014	TP DIPANGGIL

103	6B14020151	ADMIN STORE	Strata 1 Lain-lain	20/04/1989	28/01/2014	TP DIPANGGIL
104	6B14030155	2ND OFFICER TUGBOAT	SLTP dan sederajat	7/08/1976	11/03/2014	TP DIPANGGIL
106	6B14030157	A/B OIL BARGE	SLTP dan sederajat	28/10/1966	9/03/2014	TP DIPANGGIL
107	6B14030158	CHIEF OFFICER TUGBOAT	SMU Lain-lain	10/12/1972	11/03/2014	TP DIPANGGIL
108	6B14030159	2ND ENGINEER TUG BOAT	SMK Otomotif	16/08/1971	6/03/2014	TP DIPANGGIL
112	6B14030166	2ND OFFICER TUGBOAT	SMK Mesin	24/07/1987	3/03/2014	TP DIPANGGIL
115	6B14030171	CHIEF ENGINEER TUG BOAT	SMU IPS	24/08/1978	13/03/2014	TP DIPANGGIL
116	6B14030172	CHIEF OFFICER TUGBOAT	SMK Pelayaran	3/07/1991	14/03/2014	TP DIPANGGIL
118	6B14040174	SUPERVISOR CAPTAIN TUG BOAT	SMK Pelayaran	18/12/1972	25/03/2014	TP DIPANGGIL
121	6B14040181	CHIEF ENGINEER TUG BOAT	SMK Pelayaran	27/03/1972	1/04/2014	TP DIPANGGIL
122	6B14040184	SPEEDBOAT MOTORIST	SD dan sederajat	6/12/1984	25/03/2014	TP DIPANGGIL
126	6B14050190	TECHNICIAN MAINTENANCE EQUIPMENT	SMK Mesin	28/11/1986	16/04/2014	TP DIPANGGIL
127	6B14050191	CHIEF OFFICER TUGBOAT	SIM C	7/06/1983	29/04/2014	TP DIPANGGIL
128	6B14050192	CHIEF OFFICER TUGBOAT	SLTP dan sederajat	23/06/1975	30/04/2014	TP DIPANGGIL

129	6B14050193	2ND OFFICER TUGBOAT	SLTP dan sederajat	18/10/1971	25/04/2014	TP DIPANGGIL
130	6B14050194	CHIEF ENGINEER TUG BOAT	SMK Otomotif	4/02/1980	24/04/2014	TP DIPANGGIL
133	6B14050200	2ND OFFICER TUGBOAT	SMU IPS	7/04/1979	25/04/2014	TP DIPANGGIL
134	6B14050201	CHIEF ENGINEER TUG BOAT	SMU Lain-lain	12/11/1968	4/05/2014	TP DIPANGGIL
138	6B14050209	CHIEF OFFICER TUGBOAT	SMK Nautika	10/01/1984	7/05/2014	TP DIPANGGIL
139	6B14060211	COOK CRANE BARGE	KONTRAK 1 (3 BULAN)	7/02/1989	22/05/2014	TP DIPANGGIL
141	6B14070215	FOREMAN FIELD SERVICE	KTP	30/11/1982	27/06/2014	TP DIPANGGIL
147	A1	ADMIN GENERAL ADMINISTRATION & SYSTEM	Strata 1 Teknik Sipil	26/08/1987	6/02/2015	TP DIPANGGIL
148	GA1	ADMIN GENERAL ADMINISTRATION & SYSTEM	SMK Administrasi	24/12/1992	26/02/2015	TP DIPANGGIL

NIK	Tanggal lahir	Tanggungan	Jabatan
0011302564	13 Juni 1974	K2	2ND OFFICER TUGBOAT
0011303570	25 Desember 1990	TK0	2ND ENGINEER TUG BOAT
0011303572	10 Januari 1980	K1	CHIEF OFFICER TUGBOAT
0011112219	10 November 1972	K2	CAPTAIN TUG BOAT

0011304580	4 Oktober 1984	K1	SUPERINTENDENT SUPPLY CHAIN
0011305648	23 Desember 1971	K2	SUPERINTENDENT FINANCE
0011304588	10 April 1972	K1	CHIEF ENGINEER TUG BOAT
0011305650	27 Juli 1977	K2	2ND OFFICER TUGBOAT
0011305651	28 Februari 1992	TK0	A/B LCT
0011305653	7 April 1981	K2	CHIEF OFFICER TUGBOAT
0011305654	6 April 1992	TK0	MECHANIC ENGINE
0011306662	25 Desember 1973	K2	CAPTAIN TUG BOAT
0011306669	30 Juni 1979	K0	CHIEF ENGINEER TUG BOAT
0011306671	11 April 1978	K2	2ND OFFICER LCT
0011307677	23 April 1973	K3	ADMIN JETTY
0011308679	3 Desember 1978	K2	2ND OFFICER TUGBOAT
0011308680	12 November 1968	K2	2ND ENGINEER TUG BOAT
0011308681	31 Mei 1978	K2	CHIEF OFFICER TUGBOAT
0011308689	16 November 1989	K2	SPEEDBOAT MOTORIST
0011308690	23 Juni 1986	K1	ADMIN OPERATION
0011309693	26 Oktober 1975	K0	CAPTAIN TUG BOAT
0011309693	26 Oktober 1975	K0	CAPTAIN TUG BOAT
0011309695	1 November 1992	TK0	BUYER NON PART & SERVICE
0011309698	25 Juni 1976	K3	2ND OFFICER TUGBOAT
0011309699	27 Juli 1993	TK0	SPEEDBOAT MOTORIST
0011310722	22 November 1982	K2	2ND OFFICER TUGBOAT
0011310730	14 April 1990	TK0	2ND ENGINEER TUG BOAT
0011311750	7 Mei 1971	K4	CAPTAIN TUG BOAT
0011311753	7 Desember 1991	TK0	OFFICER STORE
0011311767	9 Desember 1991	TK0	OFFICER COMPENSATION AND BENEFIT
0011311757	15 Juli 1972	K3	CAPTAIN TUG BOAT
0011312760	12 Juli 1988	K1	ADMIN CLEARANCE

0011312762	24 April 1979	K0	2ND OFFICER TUGBOAT
0011312761	1 Februari 1979	K3	ENGINEER OIL BARGE
0011312763	5 Juni 1974	K1	2ND ENGINEER TUG BOAT
0011312765	20 Juni 1984	K2	CHIEF ENGINEER TUG BOAT
0011312768	21 Oktober 1987	K0	2ND ENGINEER TUG BOAT
0011401770	7 Desember 1991	TK0	TECHNICIAN ELECTRICAL
0011401776	21 September 1989	K0	OFFICER ADMINISTRATION
0011401777	14 November 1971	K3	CAPTAIN TUG BOAT
0011401780	8 Agustus 1976	K4	CHIEF ENGINEER TUG BOAT
0011401781	31 Juli 1982	K1	2ND ENGINEER TUG BOAT
0011401788	10 Agustus 1985	K2	2ND ENGINEER TUG BOAT
0011401789	9 Juni 1987	K2	2ND ENGINEER TUG BOAT
0011401787	15 Maret 1987	K1	CHIEF OFFICER TUGBOAT
0011401796	1 Agustus 1981	K0	2ND ENGINEER TUG BOAT
0011402798	15 September 1968	K4	2ND ENGINEER TUG BOAT
0011402799	5 Mei 1990	TK0	CHIEF OFFICER TUGBOAT
0011402800	13 September 1978	K2	2ND ENGINEER TUG BOAT
0011402804	21 Agustus 1989	K1	OFFICER RECRUITMENT & OD
0011403806	20 April 1989	TK0	CONTRACTING
0011403831	7 Agustus 1976	K2	2ND OFFICER TUGBOAT
0011403832	28 Oktober 1966	K3	A/B OIL BARGE
0011403816	10 Desember 1972	K2	CAPTAIN TUG BOAT
0011403817	16 Agustus 1971	K3	2ND ENGINEER TUG BOAT
0011403824	24 Juli 1987	TK0	2ND OFFICER TUGBOAT
0011403829	24 Agustus 1978	K0	CHIEF ENGINEER TUG BOAT
0011403838	3 Juli 1991	TK0	CHIEF OFFICER TUGBOAT
0011403844	18 Desember 1972	K3	CAPTAIN TUG BOAT
0011403860	27 Maret 1972	K1	CHIEF ENGINEER TUG BOAT

0011404857	6 Desember 1984	K0	SPEEDBOAT MOTORIST
0011405888	28 November 1986	K1	TECHNICIAN SUPPORT FAB. REPAIR
0011405889	7 Juni 1983	K1	CHIEF OFFICER TUGBOAT
0011405893	23 Juni 1975	K3	CHIEF OFFICER TUGBOAT
0011405884	18 Oktober 1971	K3	2ND OFFICER TUGBOAT
0011405885	4 Februari 1980	K2	CHIEF ENGINEER TUG BOAT
0011405878	7 April 1979	K2	2ND OFFICER TUGBOAT
0011405894	12 November 1968	K2	CHIEF ENGINEER TUG BOAT
0011405895	10 Januari 1984	K2	CHIEF OFFICER TUGBOAT
0011406899	7 Februari 1989	K1	COOK CRANE BARGE
0011407923	30 November 1982	TK0	FOREMAN UTV
0011502069	26 Agustus 1987	K1	ADMIN INFRASTRUCTURE & MAINTENANCE
0011503081	24 Desember 1992	TK0	ADMIN GENERAL ADMINISTRATION & SYSTEM

Jabatan Terbaru	Tanggal Diterima	Keerangan
TBJhoni XXXII	7 Februari 2013	MUTASI KE 2ND OFFICER TUGBOAT DI TBJhoni XXXII
TBJhoni V	8 Maret 2013	MUTASI KE 2ND ENGINEER TUG BOAT DI TBJhoni V
CHIEF OFFICER TUGBOAT	12 Maret 2013	PERUBAHAN STRUKTUR ORGANISASI KE CHIEF OFFICER TUGBOAT
CAPTAIN TUG BOAT	8 Desember 2011	PERUBAHAN STRUKTUR ORGANISASI KE CAPTAIN TUG BOAT
SUPERINTENDENT SUPPLY CHAIN	15 April 2013	PROMOSI JABATAN KE SUPERINTENDENT SUPPLY CHAIN
SUPERINTENDENT FINANCE	3 Mei 2013	PROMOSI JABATAN KE SUPERINTENDENT FINANCE
CHIEF ENGINEER TUG BOAT	29 April 2013	PERUBAHAN STRUKTUR ORGANISASI KE CHIEF ENGINEER TUG BOAT
TBJhoni XXXIV	4 Mei 2013	MUTASI KE 2ND OFFICER TUGBOAT DI TBJhoni XXXIV

A/B LCT	11 Mei 2013	ROTASI - PROMOSI KE A/B LCT
TBJhoni X	16 Mei 2013	MUTASI KE CHIEF OFFICER TUGBOAT DI TBJhoni X
TECHNICIAN PERIODIC SERVICE	21 Mei 2013	PERUBAHAN STRUKTUR ORGANISASI KE TECHNICIAN PERIODIC SERVICE
CAPTAIN TUG BOAT	10 Juni 2013	PERUBAHAN STRUKTUR ORGANISASI KE CAPTAIN TUG BOAT
CHIEF ENGINEER TUG BOAT	18 Juni 2013	PERUBAHAN STRUKTUR ORGANISASI KE CHIEF ENGINEER TUG BOAT
2ND OFFICER LCT	20 Juni 2013	PERUBAHAN STRUKTUR ORGANISASI KE 2ND OFFICER LCT
ADMIN PORT OPERATION	29 Juli 2013	PERUBAHAN STRUKTUR ORGANISASI KE ADMIN PORT OPERATION
2ND OFFICER TUGBOAT	14 Agustus 2013	PERUBAHAN STRUKTUR ORGANISASI KE 2ND OFFICER TUGBOAT
2ND ENGINEER TUG BOAT	14 Agustus 2013	PERUBAHAN STRUKTUR ORGANISASI KE 2ND ENGINEER TUG BOAT
TBJhoni XI	16 Agustus 2013	PERUBAHAN STRUKTUR ORGANISASI KE TBJhoni XI
SPEEDBOAT MOTORIST	22 Agustus 2013	PERUBAHAN STRUKTUR ORGANISASI KE SPEEDBOAT MOTORIST
ADMIN PORT CLEARANCE	22 Agustus 2013	PERUBAHAN STRUKTUR ORGANISASI KE ADMIN PORT CLEARANCE
CHIEF OFFICER TUGBOAT	2 September 2013	PERUBAHAN STRUKTUR ORGANISASI KE CHIEF OFFICER TUGBOAT
TBJhoni XV	2 September 2013	MUTASI KE CAPTAIN TUG BOAT DI TBJhoni XV
BUYER NON PART & SERVICE	2 September 2013	MUTASI - GROUP KE BUYER NON PART & SERVICE
2ND OFFICER TUGBOAT	12 September 2013	PERUBAHAN STRUKTUR ORGANISASI KE 2ND OFFICER TUGBOAT
SPEEDBOAT MOTORIST	12 September 2013	PERUBAHAN STRUKTUR ORGANISASI KE SPEEDBOAT MOTORIST
2ND OFFICER TUGBOAT	13 Oktober 2013	PERUBAHAN STRUKTUR ORGANISASI KE 2ND OFFICER TUGBOAT
TBJhoni XXXIV	13 Oktober 2013	MUTASI KE 2ND ENGINEER TUG BOAT DI TBJhoni XXXIV
CAPTAIN TUG BOAT	4 November 2013	PERUBAHAN STRUKTUR ORGANISASI KE CAPTAIN TUG BOAT
OFFICER STORE	11 November 2013	ROTASI - LATERAL KE OFFICER STORE
OFFICER COMPENSATION AND BENEFIT	13 November 2013	ROTASI - LATERAL KE OFFICER COMPENSATION AND BENEFIT
CAPTAIN TUG BOAT	21 November 2013	PERUBAHAN STRUKTUR ORGANISASI KE CAPTAIN TUG BOAT
ADMIN CLEARANCE	2 Desember 2013	PERUBAHAN STRUKTUR ORGANISASI KE ADMIN CLEARANCE
2ND OFFICER TUGBOAT	7 Desember 2013	PERUBAHAN STRUKTUR ORGANISASI KE 2ND OFFICER TUGBOAT

ENGINEER OIL BARGE	7 Desember 2013	PERUBAHAN STRUKTUR ORGANISASI KE ENGINEER OIL BARGE
TBJhoni II	9 Desember 2013	PERUBAHAN STRUKTUR ORGANISASI KE 2ND ENGINEER TUG BOAT
TB. JHONI XLI	13 Desember 2013	MUTASI KE CHIEF ENGINEER TUG BOAT DI TB. JHONI XLI
2ND ENGINEER TUG BOAT	23 Desember 2013	PERUBAHAN STRUKTUR ORGANISASI KE 2ND ENGINEER TUG BOAT
TECHNICIAN ELECTRICAL	6 Januari 2014	PERUBAHAN STRUKTUR ORGANISASI KE TECHNICIAN ELECTRICAL
OFFICER ADMINISTRATION	13 Januari 2014	PERUBAHAN STRUKTUR ORGANISASI KE OFFICER ADMINISTRATION
CAPTAIN TUG BOAT	17 Januari 2014	PERUBAHAN STRUKTUR ORGANISASI KE CAPTAIN TUG BOAT
CHIEF ENGINEER TUG BOAT	17 Januari 2014	PERUBAHAN STRUKTUR ORGANISASI KE CHIEF ENGINEER TUG BOAT
2ND ENGINEER TUG BOAT	17 Januari 2014	PERUBAHAN STRUKTUR ORGANISASI KE 2ND ENGINEER TUG BOAT
2ND ENGINEER TUG BOAT	17 Januari 2014	PERUBAHAN STRUKTUR ORGANISASI KE 2ND ENGINEER TUG BOAT
TBJhoni X	17 Januari 2014	MUTASI KE 2ND ENGINEER TUG BOAT DI TBJhoni X
CHIEF OFFICER TUGBOAT	17 Januari 2014	PERUBAHAN STRUKTUR ORGANISASI KE CHIEF OFFICER TUGBOAT
2ND ENGINEER TUG BOAT	23 Januari 2014	PERUBAHAN STRUKTUR ORGANISASI KE 2ND ENGINEER TUG BOAT
2ND ENGINEER TUG BOAT	2 Februari 2014	PERUBAHAN STRUKTUR ORGANISASI KE 2ND ENGINEER TUG BOAT
TBJhoni XIII	6 Februari 2014	MUTASI KE CHIEF OFFICER TUGBOAT DI TBJhoni XIII
2ND ENGINEER TUG BOAT	7 Februari 2014	PERUBAHAN STRUKTUR ORGANISASI KE 2ND ENGINEER TUG BOAT
OFFICER RECRUITMENT & OD	21 Februari 2014	PERUBAHAN STRUKTUR ORGANISASI KE OFFICER RECRUITMENT & OD
CONTRACTING	1 Maret 2014	MUTASI - GROUP KE CONTRACTING
2ND OFFICER TUGBOAT	13 Maret 2014	PERUBAHAN STRUKTUR ORGANISASI KE 2ND OFFICER TUGBOAT
A/B OIL BARGE	13 Maret 2014	PERUBAHAN STRUKTUR ORGANISASI KE A/B OIL BARGE
TBJhoni XXXVIII	14 Maret 2014	MUTASI KE CAPTAIN TUG BOAT DI TBJhoni XXXVIII
2ND ENGINEER TUG BOAT	14 Maret 2014	PERUBAHAN STRUKTUR ORGANISASI KE 2ND ENGINEER TUG BOAT
2ND OFFICER TUGBOAT	14 Maret 2014	PERUBAHAN STRUKTUR ORGANISASI KE 2ND OFFICER TUGBOAT
CHIEF ENGINEER TUG BOAT	17 Maret 2014	PERUBAHAN STRUKTUR ORGANISASI KE CHIEF ENGINEER TUG BOAT
TBJhoni IV	17 Maret 2014	MUTASI KE CHIEF OFFICER TUGBOAT DI TBJhoni IV
TB. Satria Laksana 228	29 Maret 2014	MUTASI KE CAPTAIN TUG BOAT DI TB. Satria Laksana 228

TB. Satria Laksana 228	3 April 2014	MUTASI KE CHIEF ENGINEER TUG BOAT DI TB. Satria Laksana 228
SPEEDBOAT MOTORIST	3 April 2014	PERUBAHAN STRUKTUR ORGANISASI KE SPEEDBOAT MOTORIST
TECHNICIAN SUPPORT FAB. REPAIR	1 Mei 2014	PERUBAHAN STRUKTUR ORGANISASI KE TECHNICIAN SUPPORT FAB. REPAIR
TBJhoni XV	10 Mei 2014	MUTASI KE CHIEF OFFICER TUGBOAT DI TBJhoni XV
CHIEF OFFICER TUGBOAT	10 Mei 2014	PERUBAHAN STRUKTUR ORGANISASI KE CHIEF OFFICER TUGBOAT
2ND OFFICER TUGBOAT	10 Mei 2014	PERUBAHAN STRUKTUR ORGANISASI KE 2ND OFFICER TUGBOAT
CHIEF ENGINEER TUG BOAT	10 Mei 2014	PERUBAHAN STRUKTUR ORGANISASI KE CHIEF ENGINEER TUG BOAT
2ND OFFICER TUGBOAT	10 Mei 2014	PERUBAHAN STRUKTUR ORGANISASI KE 2ND OFFICER TUGBOAT
CHIEF ENGINEER TUG BOAT	10 Mei 2014	PERUBAHAN STRUKTUR ORGANISASI KE CHIEF ENGINEER TUG BOAT
TBJhoni III	11 Mei 2014	MUTASI KE CHIEF OFFICER TUGBOAT DI TBJhoni III
COOK CRANE BARGE	3 Juni 2014	PERUBAHAN STRUKTUR ORGANISASI KE COOK CRANE BARGE
FOREMAN UTV	6 Juli 2014	PERUBAHAN STRUKTUR ORGANISASI KE FOREMAN UTV
ADMIN INFRASTRUCTURE & MAINTENANCE	10 Februari 2015	PERUBAHAN STRUKTUR ORGANISASI KE ADMIN INFRASTRUCTURE & MAINTENANCE
ADMIN GENERAL ADMINISTRATION & SYSTEM	4 Maret 2015	PERUBAHAN STRUKTUR ORGANISASI KE ADMIN GENERAL ADMINISTRATION & SYSTEM

Data Karyawan Keluar

No	No Pelamar	Posisi yang Dilamar	Pendidikan	Tanggal Lahir	Tanggal CV Diterima	Status	NIK
1	6B13010001	OFFICE BOY/GIRL BATULICIN	Diploma 2	10/06/1979	25/01/2013	TP DIPANGGIL	11302562
2	6B13020002	3RD ENGINEER TUGBOAT	SMK Lan-lain	12/05/1973	4/02/2013	TP DIPANGGIL	0011302563

20	6B13030030	3RD ENGINEER LCT	CERTIFICATE ATT-V	19/02/1981	28/03/2013	TP DIPANGGIL	0011304576
21	6B13040032	CHIEF OFFICER TUGBOAT	CERTIFICATE GOC FOR THE GMDSS	6/11/1992	6/04/2013	TP DIPANGGIL	0011304579
22	6B13040033	TECHNICIAN WELDER AND BODY REPAIR	KTP	7/01/1978	1/04/2013	TP DIPANGGIL	0011304578
23	6B13040034	OFFICER INVENTORY CONTROL	Strata 1 Biologi	29/07/1982	18/04/2013	TP DIPANGGIL	0011304582
25	6B13040037	3RD ENGINEER TUGBOAT	CERTIFICATE MEDICAL FIRST AID	23/03/1980	24/04/2013	TP DIPANGGIL	0011304585
27	6B13040040	A/B OIL BARGE	CERTIFICATE ANT-D	20/10/1990	24/04/2013	TP DIPANGGIL	0011304645
29	6B13040042	COOK OIL BARGE	CERTIFICATE ANT-D	12/09/1987	24/04/2013	TP DIPANGGIL	0011304646
34	6B13050050	2ND ENGINEER TUG BOAT	KONTRAK 1 (3 BULAN)		24/05/2013	TP DIPANGGIL	0011305657
35	6B13050051	CHIEF OFFICER TUGBOAT	SMU Lain-lain	23/06/1974	27/05/2013	TP DIPANGGIL	0011305659
36	6B13060052	CHIEF OFFICER TUGBOAT	KTP	27/02/1981	4/06/2013	TP DIPANGGIL	0011306660
37	6B13060053	3RD ENGINEER TUGBOAT	SMU Lain-lain	15/03/1983	4/06/2013	TP DIPANGGIL	0011306661
39	6B13060055	FOREMAN ELECTRICAL AND NAVIGATION	SMU IPA	3/01/1973	30/05/2013	TP DIPANGGIL	0011306667

43	6B13070060	2ND ENGINEER TUG BOAT	OIL TANKER SPECIALIZED TRAINING PROGRAMME	25/03/1970	2/07/2013	TP DIPANGGIL	0011307674
44	6B13070061	ADMIN GENERAL ADMINISTRATION & SYSTEM	Strata 1 Lain-lain	6/09/1982	28/06/2013	TP DIPANGGIL	0011307676
46	6B13080063	2ND ENGINEER TUG BOAT	SMK Pertanian	22/03/1980	28/07/2013	TP DIPANGGIL	0011308678
51	6B13080069	3RD ENGINEER TUGBOAT	SMU IPS	21/11/1984	14/08/2013	TP DIPANGGIL	0011308682
52	6B13080070	3RD ENGINEER TUGBOAT	SMK Otomotif	8/10/1982	16/08/2013	TP DIPANGGIL	0011308686
53	6B13080071	ADMIN PORT CLEARANCE	SMU IPS	12/04/1991	1/08/2013	TP DIPANGGIL	0011308691
55	6B13080073	3RD ENGINEER TUGBOAT	SMU Lain-lain	1/02/1974	26/08/2013	TP DIPANGGIL	0011308692
61	6B13090087	SUPERVISOR CAPTAIN TUG BOAT	SMU Lain-lain	6/12/1973	19/09/2013	TP DIPANGGIL	0011212518
62	6B13090088	CHIEF ENGINEER TUG BOAT	SMK Teknik	16/05/1973	19/09/2013	TP DIPANGGIL	0011309701
63	6B13090089	CHIEF OFFICER TUGBOAT	SMK Nautika	8/12/1988	5/10/2013	TP DIPANGGIL	0011310721
67	6B13090097	CHIEF OFFICER TUGBOAT	SMU IPS	10/08/1976	10/01/2013	TP DIPANGGIL	0011310729
69	6B13090099	3RD ENGINEER TUGBOAT	SMK Otomotif	16/01/1977	2/10/2013	SP DIPANGGIL	0011310731
73	6B13110107	3RD ENGINEER TUGBOAT	SMU BAHASA	6/08/1987	10/11/2013	SP DIPANGGIL	0011311756
82	6B13120118	TECHNICIAN FIELD SERVICE	SMU IPS	5/10/1992	5/12/2013	TP DIPANGGIL	0011312766
83	6B13120119	3RD ENGINEER TUGBOAT	SMK Mesin	6/05/1991	20/12/2013	TP DIPANGGIL	0011312769
88	6B14010126	CHIEF OFFICER TUGBOAT	SMU Lain-lain	17/04/1965	10/01/2014	TP DIPANGGIL	0011401778

89	6B14010127	2ND OFFICER TUGBOAT	SMU Lain-lain	22/08/1976	9/01/2014	TP DIPANGGIL	0011401779
93	6B14010131	3RD ENGINEER TUGBOAT	SMK Otomotif	26/06/1981	9/01/2014	TP DIPANGGIL	0011401782
100	6B14020146	3RD ENGINEER TUGBOAT	SMK Otomotif	18/03/1978	10/02/2014	TP DIPANGGIL	0011002081
101	6B14020148	ADMIN GENERAL ADMINISTRATION & SYSTEM	SMU Lain-lain	3/05/1985	8/02/2014	TP DIPANGGIL	0011402803
105	6B14030156	2ND ENGINEER TUG BOAT	KONTRAK 1 (3 BULAN)	26/10/1966	7/03/2014	TP DIPANGGIL	0011403830
109	6B14030160	3RD ENGINEER TUGBOAT	SMK Otomotif	10/10/1972	4/03/2014	TP DIPANGGIL	0011403818
110	6B14030164	SUPERVISOR CAPTAIN TUG BOAT	SMU IPA	20/03/1976	6/03/2014	TP DIPANGGIL	0011403822
111	6B14030165	CHIEF OFFICER TUGBOAT	SMU IPS	26/07/1979	6/03/2014	TP DIPANGGIL	0011403823
113	6B14030167	2ND ENGINEER TUG BOAT	SMK Otomotif	4/10/1974	7/03/2014	TP DIPANGGIL	0011403825
114	6B14030168	3RD ENGINEER TUGBOAT	SMK Listrik	6/12/1973	7/03/2014	TP DIPANGGIL	0011403826
117	6B14030173	CHIEF OFFICER TUGBOAT	SMK Pelayaran	11/02/1980	15/03/2014	TP DIPANGGIL	0011403839
119	6B14040175	CHIEF OFFICER TUGBOAT	SMU Lain-lain	27/05/1973	26/03/2014	TP DIPANGGIL	0011208402
120	6B14040176	3RD ENGINEER TUGBOAT	SMK Mesin	3/10/1975	26/03/2014	TP DIPANGGIL	0011403846
123	6B14040185	SPEEDBOAT MOTORIST	SD dan sederajat	13/12/1993	25/03/2014	TP DIPANGGIL	0011404856
125	6B14040188	TECHNICIAN FIELD SERVICE	SMU IPA	9/11/1984	10/04/2014	TP DIPANGGIL	0011404863
131	6B14050195	2ND ENGINEER TUG BOAT	SMK Listrik	1/01/1979	25/04/2014	TP DIPANGGIL	0011405887
132	6B14050196	3RD ENGINEER TUGBOAT	SMU Lain-lain	7/04/1989	29/04/2014	TP DIPANGGIL	0011405890

135	6B14050202	2ND ENGINEER TUG BOAT	SMU Lain-lain	14/06/1982	25/04/2014	TP DIPANGGIL	0011405883
140	6B14060212	TECHNICIAN WELDER AND BODY REPAIR	SMK Mesin	17/06/1987	26/05/2014	TP DIPANGGIL	0011406900
142	6B14070216	3RD ENGINEER TUGBOAT	SMK Pelayaran	20/12/1990	11/07/2014	TP DIPANGGIL	0011407924
144	6B14080244	2ND ENGINEER TUG BOAT	SLTP dan sederajat	20/09/1980	12/08/2014	TP DIPANGGIL	0011408975
145	6B14080245	3RD ENGINEER TUGBOAT	SMK Manajemen	12/08/1978	10/08/2014	TP DIPANGGIL	0011408976

Tanggal lahir	Tanggungan	Jabatan	Lama bekerja	Ket
10 Juni 1979	K0	OFFICE BOY/GIRL BATULICIN	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
12 Mei 1973	K0	3RD ENGINEER TUGBOAT	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
19 Februari 1981	K0	3RD ENGINEER LCT	27 Oktober 2013	VOLUNTEER-UNDUR DIRI
6 November 1992	TK0	CHIEF OFFICER TUGBOAT	29 Agustus 2013	VOLUNTEER-PHK STUDI/SERTIFIKASI
7 Januari 1978	K0	TECHNICIAN WELDER AND BODY REPAIR	2 Agustus 2013	IN VOLUNTEER-PHK KRN MANGKIR
29 Juli 1982	K0	SUPERVISOR WAREHOUSE	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
23 Maret 1980	K0	3RD ENGINEER TUGBOAT	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
20 Oktober 1990	TK0	A/B TUGBOAT	3 Januari 2016	VOLUNTEER-PHK STUDI/SERTIFIKASI
12 September 1987	K0	COOK OIL BARGE	10 Agustus 2015	VOLUNTEER-PHK STUDI/SERTIFIKASI
19 Juni 1980	K0	CHIEF ENGINEER TUG BOAT	6 Januari 2014	VOLUNTEER-UNDUR DIRI
23 Juni 1974	K0	CHIEF OFFICER TUGBOAT	2 Agustus 2016	VOLUNTEER-UNDUR DIRI
27 Februari 1981	K0	CHIEF OFFICER TUGBOAT	30 Juli 2015	IN VOLUNTEER-PHK KRN MANGKIR
15 Maret 1983	K0	3RD ENGINEER TUGBOAT	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
3 Januari 1973	K0	OFFICER ELECTRICAL & NAVIGATION	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
25 Maret 1970	K0	2ND ENGINEER TUG BOAT	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI

6 September 1982	K0	ADMIN SAFETY	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
22 Maret 1980	K0	2ND ENGINEER TUG BOAT	8 Januari 2015	VOLUNTEER-PHK STUDI/SERTIFIKASI
21 November 1984	TK0	3RD ENGINEER TUGBOAT	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
8 Oktober 1982	K0	3RD ENGINEER TUGBOAT	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
12 April 1991	K0	ADMIN OPERATION	28 April 2016	IN VOLUNTEER-PHK KRN MANGKIR
1 Februari 1974	K0	3RD ENGINEER TUGBOAT	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
6 Desember 1973	K0	SUPERVISOR CAPTAIN TUG BOAT	10 Juni 2013	VOLUNTEER-PHK STUDI/SERTIFIKASI
16 Mei 1973	K0	CHIEF ENGINEER TUG BOAT	1 Mei 2016	VOLUNTEER-UNDUR DIRI
8 Desember 1988	K0	CHIEF OFFICER TUGBOAT	31 Maret 2016	VOLUNTEER-UNDUR DIRI
8 Oktober 1976	K0	CHIEF OFFICER TUGBOAT	14 Oktober 2014	VOLUNTEER-PHK STUDI/SERTIFIKASI
16 Januari 1977	K0	3RD ENGINEER TUGBOAT	20 November 2015	VOLUNTEER-PHK STUDI/SERTIFIKASI
6 Agustus 1987	K0	3RD ENGINEER TUGBOAT	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
5 Oktober 1992	TK0	TECHNICIAN PERIODIC SERVICE	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
6 Mei 1991	TK0	3RD ENGINEER TUGBOAT	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
17 April 1965	K0	CHIEF OFFICER TUGBOAT	12 Maret 2014	VOLUNTEER-UNDUR DIRI
22 Agustus 1976	K0	2ND OFFICER TUGBOAT	19 Maret 2015	VOLUNTEER-PHK STUDI/SERTIFIKASI
26 Juni 1981	K0	3RD ENGINEER TUGBOAT	21 September 2014	VOLUNTEER-PHK KELUARGA
18 Maret 1978	K0	A/B TUGBOAT	4 Maret 2013	VOLUNTEER-PHK STUDI/SERTIFIKASI
3 Mei 1985	K0	OFFICER GENERAL ASSETS & MAINTENANCE	3 Februari 2016	VOLUNTEER-UNDUR DIRI
26 Oktober 1966	K0	2ND ENGINEER TUG BOAT	5 Juli 2014	IN VOLUNTEER-PHK KRN MANGKIR
10 Oktober 1972	TK0	3RD ENGINEER TUGBOAT	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
20 Maret 1976	K0	CAPTAIN TUG BOAT	23 Juli 2016	VOLUNTEER-UNDUR DIRI
26 Juli 1979	K0	CHIEF OFFICER TUGBOAT	2 Maret 2016	VOLUNTEER-UNDUR DIRI
4 Oktober 1974	K0	2ND ENGINEER TUG BOAT	15 April 2016	VOLUNTEER-UNDUR DIRI
6 Desember 1973	K0	2ND ENGINEER TUG BOAT	11 Maret 2016	VOLUNTEER-UNDUR DIRI
11 Februari 1980	K0	CHIEF OFFICER TUGBOAT	30 Desember 2015	VOLUNTEER-UNDUR DIRI
27 Mei 1973	K0	CHIEF OFFICER TUGBOAT	16 November 2013	VOLUNTEER-UNDUR DIRI
3 Oktober 1975	K0	3RD ENGINEER TUGBOAT	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
13 Desember 1993	TK0	SPEEDBOAT MOTORIST	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
9 November 1984	K0	ADMIN	5 Oktober 2014	IN VOLUNTEER-PHK KRN MANGKIR
1 Januari 1979	K0	2ND ENGINEER TUG BOAT	19 Januari 2015	VOLUNTEER-PHK STUDI/SERTIFIKASI

7 April 1989	TK0	3RD ENGINEER TUGBOAT	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
14 Juni 1982	K0	2ND ENGINEER TUG BOAT	5 Agustus 2016	VOLUNTEER-UNDUR DIRI
17 Juni 1987	TK0	TECHNICIAN ENGINEERING & QC	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI
20 Desember 1990	TK0	2ND ENGINEER TUG BOAT	1 Februari 2016	IN VOLUNTEER-PHK KRN MANGKIR
20 September 1980	K0	2ND ENGINEER TUG BOAT	7 April 2016	VOLUNTEER-UNDUR DIRI
12 Agustus 1978	K0	3RD ENGINEER TUGBOAT	29 Februari 2016	IN VOLUNTEER-PHK RASIONALISASI

LAMPIRAN B

Data test arff

```
@relation penerimaanpegawai

@attribute posisi {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31}
@attribute didik {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40}
@attribute usia numeric
@attribute tanggung {1,2,3,4,5,6}
@attribute stat {1,2}
@attribute ket {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49}
@attribute class {0,1}
```

```
@data
3,7,49,4,1,46,0
1,7,37,4,1,47,0
15,8,33,4,1,49,0
15,8,36,4,1,47,0
16,7,29,4,1,46,0
2,9,47,4,1,48,0
15,8,41,2,1,45,0
10,35,38,4,1,46,0
3,15,35,4,1,46,0
2,8,39,4,1,46,0
15,39,40,4,1,46,0
3,20,34,4,1,46,0
3,7,41,4,1,46,0
15,14,39,4,1,45,0
14,24,20,2,1,45,0
31,35,29,4,1,48,0
2,39,35,4,1,47,0
15,7,25,2,1,45,0
2,7,32,4,1,46,0
27,14,27,2,1,45,0
15,20,24,2,1,48,0
2,10,34,4,1,46,0
15,40,36,4,1,45,0
21,23,25,2,1,32,1
1,10,38,1,1,15,1
```

```
1,10,38,1,1,15,1
7,10,47,5,1,33,1
3,7,41,1,1,34,1
2,8,43,5,1,16,1
1,14,27,2,1,15,1
11,15,36,4,1,7,1
3,20,23,2,1,35,1
10,20,41,5,1,36,1
11,20,42,3,1,37,1
14,24,29,4,1,18,1
22,14,27,3,1,38,1
3,25,31,3,1,39,1
3,10,39,5,1,3,1
1,10,43,5,1,15,1
11,8,34,1,1,7,1
1,15,35,1,1,15,1
11,7,46,1,1,7,1
3,13,30,1,1,40,1
23,9,25,3,1,41,1
24,21,32,2,1,42,1
16,27,27,3,1,43,1
16,28,22,2,1,44,1
```

Data Train Arff

```

@relation penerimaanpegawai

@attribute posisi {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31}
@attribute didik {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40}
@attribute usia numeric
@attribute tanggung {1,2,3,4,5,6}
@attribute stat {1,2}
@attribute ket {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49}
@attribute class {0,1}

@data
1,1,39,1,1,1,1
2,2,22,2,1,2,1
3,3,33,3,1,3,1
1,4,32,1,2,4,1
4,5,29,3,1,5,1
5,5,41,1,1,6,1
6,2,41,3,1,7,1
7,6,36,1,1,8,1
8,7,21,2,1,9,1
3,6,32,1,1,10,1
9,8,21,2,1,12,1
10,9,39,1,2,4,1
11,7,34,4,1,7,1
12,7,35,1,2,13,1
13,10,40,5,1,14,1
1,11,35,1,1,15,1
2,12,45,1,1,16,1
3,13,35,1,1,17,1
14,7,24,1,1,18,1
13,14,27,3,1,19,1
3,9,38,4,1,3,1
15,15,38,4,1,20,1
16,16,21,2,1,21,1
1,15,37,5,1,15,1
14,8,20,2,1,18,1
1,15,31,1,1,15,1
2,11,23,2,1,22,1
10,13,43,6,2,4,1
17,5,22,2,2,23,1
18,17,22,2,2,24,1
3,18,41,5,2,4,1
13,15,25,3,2,25,1
1,10,35,4,2,15,1
6,15,35,5,2,26,1
2,19,40,3,2,16,1
11,20,29,1,1,27,1
2,8,26,4,1,16,1
19,15,22,2,1,28,1
20,7,24,4,1,29,1
10,13,42,5,1,4,1
11,9,37,6,1,7,1
2,12,31,3,1,16,1
2,8,28,1,1,16,1
15,8,27,1,1,30,1
3,20,27,3,1,3,1
2,11,32,4,1,16,1
2,7,45,6,1,16,1
3,13,24,2,1,11,1
2,21,35,1,1,16,1
16,22,24,3,1,31,1
25,29,34,4,1,45,0
15,16,40,4,1,45,0
26,30,32,4,1,46,0
3,31,20,2,1,47,0
27,21,35,4,1,48,0
28,32,31,4,1,45,0
15,33,33,4,1,45,0
7,34,23,2,1,47,0
29,34,26,4,1,47,0
2,9,33,4,1,46,0
3,7,39,4,1,46,0
3,21,32,4,1,48,0
15,7,30,4,1,45,0
30,35,40,4,1,45,0
2,36,43,4,1,45,0
16,23,31,4,1,45,0
2,37,33,4,1,47,0
15,15,29,2,1,45,0
15,8,31,4,1,45,0
13,15,22,4,1,48,0
15,7,40,4,1,45,0
10,7,40,4,1,47,0
11,11,40,4,1,46,0
3,13,25,4,1,46,0
3,15,36,4,1,47,0
15,8,37,4,2,47,0
15,38,26,4,2,45,0
31,15,21,2,1,45,0
15,14,23,2,1,45,0

```

LAMPIRAN C

Source Code

1. Kelas Utama

```
import weka.classifiers.Evaluation;
import weka.classifiers.trees.RandomForest;
import weka.core.Instances;
import java.util.Arrays;
import java.util.Random;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
import weka.core.converters.ConverterUtils.DataSource;

public class GuiRandomForest extends javax.swing.JFrame {

    private String end;

    public static class chromosomeData implements
Comparable<chromosomeData> {
        boolean[] chromosome;
        double accuracy;

        public chromosomeData(boolean[] data,double acc)
        {
            this.chromosome = data;
            this.accuracy = acc;
        }
        @Override
        public int compareTo(chromosomeData o) {
            return (int) (o.accuracy - this.accuracy);
        }
    }
}
```

```

public static boolean[] integer_to_chromosome(int temp1)
{
    boolean[] bits = new boolean[6];
    int temp=temp1;

    for (int i = 5; i >= 0; i--) {
        bits[5-i] = (temp & (1 << i)) != 0;
    }

    return bits;
}

public static Instances chromosome(Instances input,
boolean bits[])
{
    Instances temp=input;

    if (temp.classIndex() == -1)
        temp.setClassIndex(temp.numAttributes() - 1);

    for(int i=5;i>=0;i--)
    {
        if(bits[i]==false)
        {
            temp.deleteAttributeAt(i);
        }
    }
    return temp;
}

public static double doRandomForest(Instances training,
Instances testing) throws Exception
{
    double accuracy;

```

```

String[] options = new String[1];
options[0] = "-U";
RandomForest tree = new RandomForest();
tree.setOptions(options);
tree.buildClassifier(training);

Evaluation eval = new Evaluation(testing);
eval.evaluateModel(tree, testing);
accuracy = (eval.correct()/56)*100;

return accuracy;
}

public void RandomForest() throws Exception {
    DataSource source = new
DataSource("/kuliah/TA/fix/randomforest/src/randomforest/tr
ain.arff");
    Instances training = source.getDataSet();
    if (training.classIndex() == -1)
        training.setClassIndex(training.numAttributes() - 1);

    DataSource testing = new
DataSource("/kuliah/TA/fix/randomforest/src/randomforest/te
st.arff");
    Instances test = testing.getDataSet();
    if (test.classIndex() == -1)
        test.setClassIndex(test.numAttributes() - 1);

    int max = 64;
    int min = 1;
    int i;
    Random r = new Random();
    int randInt;

```



```

int input;

boolean[] inputbits = new boolean[6];
Instances inputTrain;
Instances inputTest;

double acc = 0;
chromosomeData[] data = new chromosomeData[200];

for(i=0;i<200;i++)
{
    input = 0;
    chromosomeData tempData = new
chromosomeData(inputbits,acc);
    data[i] = tempData;
}

System.out.println("-----Initialize initial chromosome---
---");

Scanner sc = new Scanner(System.in);
System.out.print("Masukkan jumlah inisial kromosom: ");
int a = Integer.parseInt(jTextField1.getText());
for(i=0;i<a;i++)

{
    randInt = r.nextInt(max-min) + min;
    input = randInt;
    inputbits = integer_to_chromosome(input);
    inputTrain =
chromosome(source.getDataSet(),inputbits);
    inputTest = chromosome(testing.getDataSet(),inputbits);
    acc = doRandomForest(inputTrain,inputTest);
    chromosomeData tempData = new
chromosomeData(inputbits,acc);

```

```

data[i] = tempData;

System.out.println("kromosom = " +
Arrays.toString(inputbits)+" accuracy = "+acc);
Arrays.sort(data);

}

System.out.println("-----sorted-----");
System.out.println("Sort hasil random forest");
String inisial = "";
for(i=0;i<a;i++)

{
System.out.println("kromosom = " +
Arrays.toString(data[i].chromosome)+" acc =
"+data[i].accuracy);

    inisial = inisial + ("kromosom = " +
Arrays.toString(data[i].chromosome)+" acc =
"+data[i].accuracy+"\n");
    jTextArea2.setText(inisial);
}
System.out.println("-----do Genetic Algorithm with
random forest-----");
boolean[] chromosome1 = new boolean[6];
boolean[] chromosome2 = new boolean[6];
int maxchromosome = 11;
int minchromosome = 1;
Random rchromosome = new Random();
int index;
int point;

int init=19;

```

```

        for(i=0;i<30;i++)
        {
            System.out.println("-----GA ke-" + i + " -----");
            randInt = rchromosome.nextInt(maxchromosome-
minchromosome) + minchromosome;
            index = randInt;
            chromosome1 = data[index].chromosome;
            randInt = rchromosome.nextInt(maxchromosome-
minchromosome) + minchromosome;
            index = randInt;
            chromosome2 = data[index].chromosome;
            randInt = rchromosome.nextInt(6-2) + 2;
            point = randInt;
            chromosome1 =
crossover1(chromosome1,chromosome2,point);
            chromosome2 =
crossover2(chromosome1,chromosome2,point);
            chromosome1 = mutation(chromosome1);
            chromosome2 = mutation(chromosome2);

            inputTrain =
chromosome(source.getDataSet(),chromosome1);
            inputTest =
chromosome(testing.getDataSet(),chromosome1);
            acc = doRandomForest(inputTrain,inputTest);
            chromosomeData tempData = new
chromosomeData(chromosome1,acc);
            init=init+1;
            data[init] = tempData;
            System.out.println("kromosom1 = " +
Arrays.toString(chromosome1)+" accuracy = "+acc);
            inputTrain =
chromosome(source.getDataSet(),chromosome2);
            inputTest =
chromosome(testing.getDataSet(),chromosome2);

```

```

        acc = doRandomForest(inputTrain,inputTest);
        chromosomeData tempData2 = new
chromosomeData(chromosome2,acc);
        init=init+1;
        data[init] = tempData2;

        System.out.println("kromosom2 = " +
Arrays.toString(chromosome2)+" accuracy = "+acc);

        Arrays.sort(data);
        System.out.println("-----sorted-----");

    }

    System.out.println("-----Urutan kromosom terbaik-----
-");
    end = "";
    for(i=0;i<a;i++)
    {
        System.out.println("kromosom = " +
Arrays.toString(data[i].chromosome)+" acc =
"+data[i].accuracy);
        end = end + ("kromosom = " +
Arrays.toString(data[i].chromosome)+" acc =
"+data[i].accuracy+"\n");
    }

}

```

```

private void initComponents() {

    jLabel2 = new javax.swing.JLabel();
    jPanel1 = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTextArea1 = new javax.swing.JTextArea();
    jScrollPane2 = new javax.swing.JScrollPane();
    jTextArea2 = new javax.swing.JTextArea();
    jLabel1 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jTextField1 = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();
    jLabel4 = new javax.swing.JLabel();

    jLabel2.setFont(new java.awt.Font("Tahoma", 0, 18)); //
    NOI18N
    jLabel2.setText("Inisial Awal Kromosom Random
    Forest");

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jTextArea1.setColumns(20);
    jTextArea1.setRows(5);
    jScrollPane1.setViewportView(jTextArea1);

    jTextArea2.setColumns(20);
    jTextArea2.setRows(5);
    jScrollPane2.setViewportView(jTextArea2);

```

```

        jLabel1.setFont(new java.awt.Font("Tahoma", 0, 18)); //
        NOI18N
        jLabel1.setText("Inisial Awal Kromosom Random
        Forest");

        jLabel3.setFont(new java.awt.Font("Tahoma", 0, 18)); //
        NOI18N
        jLabel3.setText("Urutan Kromosom Terbaik");

        jTextField1.addActionListener(new
        java.awt.event.ActionListener() {
            public void
            actionPerformed(java.awt.event.ActionEvent evt) {
                jTextField1ActionPerformed(evt);
            }
        });

        jButton1.setLabel("Start Random Forest");
        jButton1.addActionListener(new
        java.awt.event.ActionListener() {
            public void
            actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setLabel("Start Genetic Algorithm");
        jButton2.addActionListener(new
        java.awt.event.ActionListener() {
            public void
            actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

```

```

        jLabel4.setFont(new java.awt.Font("Tahoma", 0, 12)); //
        NOI18N
        jLabel4.setText("Inisial Awal Kromosom:");

        javax.swing.GroupLayout jPanel1Layout = new
        javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(29, 29, 29)

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.G
        roupLayout.Alignment.LEADING)

            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(29, 29, 29)

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.G
        roupLayout.Alignment.TRAILING)
                .addGroup(jPanel1Layout.createSequentialGroup()

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.G
        roupLayout.Alignment.TRAILING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(29, 29, 29)
                    .addComponent(jButton2,
        javax.swing.GroupLayout.DEFAULT_SIZE, 189,
        Short.MAX_VALUE)

            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(29, 29, 29)
                .addComponent(jScrollPane1,
        javax.swing.GroupLayout.PREFERRED_SIZE, 420,
        javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(29, 29, 29)
                .addComponent(jButton2,
        javax.swing.GroupLayout.DEFAULT_SIZE, 189,
        Short.MAX_VALUE)
    
```

```

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.G
roupLayout.Alignment.TRAILING)

.addGroup(jPanel1Layout.createSequentialGroup()
    .addComponent(jLabel4,
javax.swing.GroupLayout.PREFERRED_SIZE, 132,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlace
ment.RELATED)
    .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(18, 18, 18))

.addGroup(jPanel1Layout.createSequentialGroup()
    .addComponent(jButton1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    .addGap(6, 6, 6))
    .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 420,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addContainerGap())

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
    .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 316,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(64, 64, 64))

```



```

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
    .addComponent(jLabel3)
    .addGap(118, 118, 118))))
);
jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
    .addContainerGap(29, Short.MAX_VALUE)
    .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 21,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlace
ment.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.G
roupLayout.Alignment.LEADING)
    .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 118,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel1Layout.createSequentialGroup())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.G
roupLayout.Alignment.BASELINE)
    .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(jLabel4,
        javax.swing.GroupLayout.PREFERRED_SIZE, 21,
        javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlace
        ment.RELATED)

        .addComponent(jButton1,
        javax.swing.GroupLayout.PREFERRED_SIZE, 53,
        javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlace
        ment.RELATED)

        .addComponent(jLabel3,
        javax.swing.GroupLayout.PREFERRED_SIZE, 21,
        javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlace
        ment.RELATED)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.G
        roupLayout.Alignment.LEADING)

        .addGroup(jPanel1Layout.createSequentialGroup()
        .addComponent(jScrollPane1,
        javax.swing.GroupLayout.PREFERRED_SIZE, 118,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(55, 55, 55))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
        jPanel1Layout.createSequentialGroup()
        .addComponent(jButton2,
        javax.swing.GroupLayout.PREFERRED_SIZE, 53,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(90, 90, 90))))

    );

```

```

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
            .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
            .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        );

        pack();
    } // </editor-fold>

    private void
jTextField1ActionPerformed(java.awt.event.ActionEvent evt)
    {
        // TODO add your handling code here:
    }

    private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

```

```

        try {
            // TODO add your handling code here:
            RandomForest();

        } catch (Exception ex) {

Logger.getLogger(GuiRandomForest.class.getName()).log(L
evel.SEVERE, null, ex);
        }
    }

    private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try {
            // TODO add your handling code here:

            jTextArea1.setText(end);
        } catch (Exception ex) {

Logger.getLogger(GuiRandomForest.class.getName()).log(L
evel.SEVERE, null, ex);
        }
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and
        feel setting code (optional) ">

```

```

    /* If Nimbus (introduced in Java SE 6) is not available,
    stay with the default look and feel.
    * For details see
    http://download.oracle.com/javase/tutorial/uiswing/lookandfe
    el/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName(
                ));
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(GuiRandomForest.class.g
        etName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(GuiRandomForest.class.g
        etName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(GuiRandomForest.class.g
        etName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch
    (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(GuiRandomForest.class.g
        etName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

```

```

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new GuiRandomForest().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTextArea jTextArea1;
private javax.swing.JTextArea jTextArea2;
private javax.swing.JTextField jTextField1;
// End of variables declaration
}

```

2. Crossover

```

public static boolean[] crossover1(boolean bits1[],boolean
bits2[],int point)
{
    boolean[] bits = new boolean[6];

    int i;

    for(i=0;i<6;i++)

```

```
{
    if(i<point){
        bits[i]=bits1[i];
    }
    else
    {
        bits[i]=bits2[i];
    }
}
return bits;
}

public static boolean[] crossover2(boolean bits1[],boolean
bits2[],int point)
{
    boolean[] bits = new boolean[6];

    int i;

    for(i=0;i<6;i++)
    {
        if(i<point){
            bits[i]=bits2[i];
        }
        else
        {
            bits[i]=bits1[i];
        }
    }
    return bits;
}
```

3. Mutasi

```
public static boolean[] mutation(boolean input[])
{
    boolean[] bits = new boolean[6];

    int max =2;
    int min= 0;
    int i;
    Random r = new Random();
    int randInt;

    for(i=0;i<6;i++)
    {
        randInt = r.nextInt(max-min) + min;

        if(randInt==0){
            if(input[i]==false)
            {
                bits[i]=true;
            }
            else
            {
                bits[i]=false;
            }
        }

        else
        {
            bits[i] = input[i];
        }

    }
}
```



```
return bits;  
}
```

LAMPIRAN D

Uji Coba 1

Random Forests:

1. kromosom = [true, false, true, true, true, true] acc = 95.65217391304348
2. kromosom = [true, true, true, true, false, true] acc = 95.65217391304348
3. kromosom = [true, true, true, true, true, true] acc = 93.47826086956522
4. kromosom = [true, true, true, false, true, true] acc = 93.47826086956522
5. kromosom = [true, false, true, true, false, true] acc = 93.47826086956522
6. kromosom = [true, true, false, false, false, true] acc = 91.30434782608695
7. kromosom = [true, false, true, true, true, false] acc = 89.13043478260869
8. kromosom = [false, false, false, true, true, true] acc = 89.13043478260869
9. kromosom = [true, true, false, false, true, true] acc = 86.95652173913044
10. kromosom = [false, true, false, false, true, true] acc = 82.6086956521739
11. kromosom = [true, true, false, true, true, false] acc = 82.6086956521739
12. kromosom = [false, true, false, false, true, true] acc = 82.6086956521739
13. kromosom = [false, false, true, false, true, true] acc = 82.6086956521739
14. kromosom = [false, false, true, true, true, false] acc = 78.26086956521739

15. kromosom = [false, false, false, true, false, true] acc = 69.56521739130434
16. kromosom = [true, true, false, false, true, false] acc = 69.56521739130434
17. kromosom = [false, false, false, true, false, true] acc = 69.56521739130434
18. kromosom = [false, false, true, false, false, false] acc = 45.65217391304348
19. kromosom = [false, false, true, false, true, false] acc = 43.47826086956522
20. kromosom = [false, false, true, false, true, false] acc = 43.47826086956522

Setelah dioptimalkan dengan Algoritma Genetika:

1. kromosom = [true, false, true, true, true, true] acc = 95.65217391304348
2. kromosom = [true, true, true, true, false, true] acc = 95.65217391304348
3. kromosom = [true, false, true, true, true, true] acc = 95.65217391304348
4. kromosom = [true, false, true, true, true, true] acc = 95.65217391304348
5. kromosom = [true, true, true, true, true, true] acc = 93.47826086956522
6. kromosom = [true, true, true, false, true, true] acc = 93.47826086956522
7. kromosom = [true, false, true, true, false, true] acc = 93.47826086956522
8. kromosom = [false, true, false, true, false, true] acc = 93.47826086956522
9. kromosom = [true, true, false, true, false, true] acc = 93.47826086956522
10. kromosom = [true, true, true, true, true, true] acc = 93.47826086956522

11. kromosom = [false, true, false, true, false, true] acc = 93.47826086956522
12. kromosom = [true, true, true, true, true, true] acc = 93.47826086956522
13. kromosom = [false, true, false, true, false, true] acc = 93.47826086956522
14. kromosom = [true, false, true, true, false, true] acc = 93.47826086956522
15. kromosom = [true, true, false, true, true, true] acc = 93.47826086956522
16. kromosom = [false, true, true, true, true, true] acc = 93.47826086956522
17. kromosom = [true, true, true, false, true, true] acc = 93.47826086956522
18. kromosom = [true, true, false, true, true, true] acc = 93.47826086956522
19. kromosom = [true, true, true, true, true, true] acc = 93.47826086956522
20. kromosom = [true, true, false, false, false, true] acc = 91.30434782608695

Uji Coba 2

Random Forest:

1. kromosom = [true, false, true, true, false, true] acc = 93.47826086956522
2. kromosom = [true, true, true, false, true, true] acc = 93.47826086956522
3. kromosom = [true, true, true, false, true, true] acc = 93.47826086956522
4. kromosom = [true, true, true, true, true, true] acc = 93.47826086956522
5. kromosom = [false, false, false, true, true, true] acc = 89.13043478260869

6. kromosom = [false, false, false, true, true, true] acc = 89.13043478260869
7. kromosom = [true, false, true, true, false, false] acc = 89.13043478260869
8. kromosom = [true, true, false, false, true, true] acc = 86.95652173913044
9. kromosom = [true, true, false, true, false, false] acc = 82.6086956521739
10. kromosom = [true, true, true, true, false, false] acc = 78.26086956521739
11. kromosom = [true, false, false, false, false, true] acc = 69.56521739130434
12. kromosom = [false, false, false, false, true, true] acc = 69.56521739130434
13. kromosom = [false, false, false, true, false, true] acc = 69.56521739130434
14. kromosom = [true, true, true, false, false, false] acc = 69.56521739130434
15. kromosom = [false, true, false, true, true, false] acc = 67.3913043478261
16. kromosom = [true, false, false, false, true, false] acc = 60.86956521739131
17. kromosom = [true, false, false, false, false, false] acc = 60.86956521739131
18. kromosom = [false, true, true, false, true, false] acc = 54.347826086956516
19. kromosom = [false, false, true, false, false, false] acc = 45.65217391304348
20. kromosom = [false, true, false, false, false, false] acc = 41.30434782608695

Setelah dioptimalkan dengan Algoritma Genetika:

1. kromosom = [true, false, false, true, false, true] acc = 95.65217391304348
2. kromosom = [true, false, true, true, true, true] acc = 95.65217391304348
3. kromosom = [true, false, false, true, false, true] acc = 95.65217391304348
4. kromosom = [true, false, false, true, true, true] acc = 95.65217391304348
5. kromosom = [true, false, true, true, false, true] acc = 93.47826086956522
6. kromosom = [true, true, true, false, true, true] acc = 93.47826086956522
7. kromosom = [true, true, true, false, true, true] acc = 93.47826086956522
8. kromosom = [true, true, true, true, true, true] acc = 93.47826086956522
9. kromosom = [true, true, true, true, true, true] acc = 93.47826086956522
10. kromosom = [true, true, true, false, true, true] acc = 93.47826086956522
11. kromosom = [false, true, false, true, false, true] acc = 93.47826086956522
12. kromosom = [false, true, true, true, true, true] acc = 93.47826086956522
13. kromosom = [true, true, true, false, false, true] acc = 91.30434782608695
14. kromosom = [false, false, false, true, true, true] acc = 89.13043478260869
15. kromosom = [false, false, false, true, true, true] acc = 89.13043478260869
16. kromosom = [true, false, true, true, false, false] acc = 89.13043478260869
17. kromosom = [true, false, false, true, true, false] acc = 89.13043478260869

18. kromosom = [false, false, false, true, true, true] acc = 89.13043478260869
19. kromosom = [true, false, true, true, true, false] acc = 89.13043478260869
20. kromosom = [false, false, false, true, true, true] acc = 89.13043478260869

Uji Coba 3

Random Forest:

1. kromosom = [true, true, true, false, true, true] acc = 93.47826086956522
2. kromosom = [true, false, false, true, false, false] acc = 89.13043478260869
3. kromosom = [false, false, true, true, true, true] acc = 89.13043478260869
4. kromosom = [false, true, true, false, false, true] acc = 84.78260869565217
5. kromosom = [true, false, false, false, true, true] acc = 80.43478260869566
6. kromosom = [false, false, true, true, true, false] acc = 78.26086956521739
7. kromosom = [true, true, true, false, true, false] acc = 69.56521739130434
8. kromosom = [false, false, false, false, false, true] acc = 69.56521739130434
9. kromosom = [true, true, false, false, true, false] acc = 69.56521739130434
10. kromosom = [false, false, true, false, false, false] acc = 45.65217391304348

Setelah dioptimalkan dengan Algoritma Genetika:

1. kromosom = [true, false, false, true, true, true] acc = 95.65217391304348

2. kromosom = [true, false, false, true, true, true] acc = 95.65217391304348
3. kromosom = [true, false, false, true, true, true] acc = 95.65217391304348
4. kromosom = [true, false, true, true, true, true] acc = 95.65217391304348
5. kromosom = [true, true, true, true, false, true] acc = 95.65217391304348
6. kromosom = [true, false, false, true, false, true] acc = 95.65217391304348
7. kromosom = [true, true, true, true, false, true] acc = 95.65217391304348
8. kromosom = [true, true, true, false, true, true] acc = 93.47826086956522
9. kromosom = [true, true, true, false, true, true] acc = 93.47826086956522
10. kromosom = [false, true, true, true, false, true] acc = 93.47826086956522

Uji Coba 4

Random Forest:

1. kromosom = [true, false, false, true, true, true] acc = 95.65217391304348
2. kromosom = [true, true, false, true, true, true] acc = 93.47826086956522
3. kromosom = [false, true, true, true, true, true] acc = 93.47826086956522
4. kromosom = [false, true, false, true, false, true] acc = 93.47826086956522
5. kromosom = [false, true, false, true, true, true] acc = 91.30434782608695
6. kromosom = [true, true, false, false, false, true] acc = 91.30434782608695

7. kromosom = [true, false, true, true, false, false] acc = 89.13043478260869
8. kromosom = [true, false, true, true, false, false] acc = 89.13043478260869
9. kromosom = [false, true, true, false, false, true] acc = 84.78260869565217
10. kromosom = [false, true, true, false, false, true] acc = 84.78260869565217
11. kromosom = [false, false, false, true, false, false] acc = 84.78260869565217
12. kromosom = [true, true, false, true, false, false] acc = 82.6086956521739
13. kromosom = [true, true, false, true, false, false] acc = 82.6086956521739
14. kromosom = [true, true, true, false, true, false] acc = 69.56521739130434
15. kromosom = [true, true, false, false, true, false] acc = 69.56521739130434
16. kromosom = [true, false, true, false, false, false] acc = 56.52173913043478
17. kromosom = [false, false, true, false, false, false] acc = 45.65217391304348

Setelah dioptimalkan dengan Algoritma Genetika:

1. kromosom = [true, false, false, true, true, true] acc = 95.65217391304348
2. kromosom = [true, false, true, true, true, true] acc = 95.65217391304348
3. kromosom = [true, false, false, true, false, true] acc = 95.65217391304348
4. kromosom = [true, false, false, true, true, true] acc = 95.65217391304348
5. kromosom = [true, true, false, true, true, true] acc = 93.47826086956522

6. kromosom = [false, true, true, true, true, true] acc = 93.47826086956522
7. kromosom = [false, true, false, true, false, true] acc = 93.47826086956522
8. kromosom = [true, true, true, true, true, true] acc = 93.47826086956522
9. kromosom = [true, true, false, true, true, true] acc = 93.47826086956522
10. kromosom = [false, true, true, true, false, true] acc = 93.47826086956522
11. kromosom = [true, false, true, true, false, true] acc = 93.47826086956522
12. kromosom = [false, true, false, true, true, true] acc = 91.30434782608695
13. kromosom = [true, true, false, false, false, true] acc = 91.30434782608695
14. kromosom = [false, false, true, true, false, true] acc = 91.30434782608695
15. kromosom = [true, true, true, false, false, true] acc = 91.30434782608695
16. kromosom = [false, false, true, true, false, true] acc = 91.30434782608695
17. kromosom = [false, false, true, true, false, true] acc = 91.30434782608695

Uji Coba 5

Random Forest:

1. kromosom = [true, true, true, true, false, true] acc = 95.65217391304348
2. kromosom = [true, false, false, true, false, true] acc = 95.65217391304348
3. kromosom = [true, false, true, true, false, true] acc = 93.47826086956522

4. kromosom = [true, false, true, true, false, true] acc = 93.47826086956522
5. kromosom = [false, false, true, true, true, true] acc = 89.13043478260869
6. kromosom = [false, false, false, true, true, true] acc = 89.13043478260869
7. kromosom = [true, false, true, true, false, false] acc = 89.13043478260869
8. kromosom = [false, false, false, true, true, true] acc = 89.13043478260869
9. kromosom = [true, false, true, true, false, false] acc = 89.13043478260869
10. kromosom = [true, true, false, false, true, true] acc = 86.95652173913044
11. kromosom = [false, true, true, false, true, true] acc = 84.78260869565217
12. kromosom = [false, true, true, false, false, true] acc = 84.78260869565217
13. kromosom = [true, false, true, false, false, true] acc = 80.43478260869566
14. kromosom = [true, true, true, true, false, false] acc = 78.26086956521739
15. kromosom = [false, false, true, true, true, false] acc = 78.26086956521739
16. kromosom = [false, false, true, true, false, false] acc = 76.08695652173914
17. kromosom = [true, true, false, false, false, false] acc = 71.73913043478261
18. kromosom = [true, false, true, false, true, false] acc = 71.73913043478261
19. kromosom = [false, false, false, true, false, true] acc = 69.56521739130434
20. kromosom = [false, true, false, false, false, true] acc = 69.56521739130434

21. kromosom = [false, true, false, true, false, false] acc = 67.3913043478261
22. kromosom = [false, false, true, false, true, false] acc = 43.47826086956522
23. kromosom = [false, true, false, false, false, false] acc = 41.30434782608695

Setelah dioptimalkan dengan Algoritma Genetika:

1. kromosom = [true, true, true, true, false, true] acc = 95.65217391304348
2. kromosom = [true, false, false, true, false, true] acc = 95.65217391304348
3. kromosom = [true, false, false, true, false, true] acc = 95.65217391304348
4. kromosom = [true, false, true, true, true, true] acc = 95.65217391304348
5. kromosom = [true, false, false, true, true, true] acc = 95.65217391304348
6. kromosom = [true, false, true, true, false, true] acc = 93.47826086956522
7. kromosom = [true, false, true, true, false, true] acc = 93.47826086956522
8. kromosom = [true, true, true, true, true, true] acc = 93.47826086956522
9. kromosom = [false, true, false, true, false, true] acc = 93.47826086956522
10. kromosom = [false, true, true, true, false, true] acc = 93.47826086956522
11. kromosom = [true, true, true, false, true, true] acc = 93.47826086956522
12. kromosom = [true, true, false, true, true, true] acc = 93.47826086956522
13. kromosom = [false, true, true, true, true, true] acc = 93.47826086956522

14. kromosom = [false, true, false, true, false, true] acc = 93.47826086956522
15. kromosom = [false, true, false, true, true, true] acc = 91.30434782608695
16. kromosom = [true, true, true, false, false, true] acc = 91.30434782608695
17. kromosom = [true, true, true, false, false, true] acc = 91.30434782608695
18. kromosom = [false, false, true, true, true, true] acc = 89.13043478260869
19. kromosom = [false, false, false, true, true, true] acc = 89.13043478260869
20. kromosom = [true, false, true, true, false, false] acc = 89.13043478260869
21. kromosom = [false, false, false, true, true, true] acc = 89.13043478260869
22. kromosom = [true, false, true, true, false, false] acc = 89.13043478260869
23. kromosom = [true, false, true, true, false, false] acc = 89.13043478260869

LAMPIRAN E

BIODATA PENULIS



Penulis bernama Laras Binarwati atau yang akrab disapa Laras, lahir di Bontang, 30 Desember 1993. Penulis merupakan anak ketiga pasangan Nurcholis dan Agustina. Penulis menempuh pendidikan formal di SD II Yayasan Pupuk Kaltim Bontang (2000-2006), SMP Yayasan Pupuk Kaltim Bontang (2006-2009), SMA Yayasan Pupuk Kaltim Bontang (2009-2012). Setelah lulus dari SMA, penulis melanjutkan studi ke jenjang S1 di Jurusan Matematika ITS Surabaya pada tahun 2012. Di Jurusan Matematika, penulis mengambil Bidang Minat Ilmu Komputer. Selain aktif kuliah, penulis juga aktif mengikuti organisasi, di antaranya: Sekertaris Dalam Negeri HIMATIKA ITS (2013-2014). Tidak hanya itu, Penulis juga aktif berperan dalam beberapa kegiatan kepanitiaan, seperti: Olimpiade Matematika ITS (2013-2015), LKMM Pra TD (2014), LKMM TD (2014) dan Gerigi ITS (2014).

Adapun saran, kritik, dan pertanyaan mengenai Tugas Akhir ini, dapat ditujukan melalui email penulis: larasbinarwati@gmail.com